Mémoire présenté le :
pour l'obtention du diplôme
de Statisticien Mention Actuariat
et l'admission à l'Institut des Actuaires

Par : Léa COCHET

Titre du mémoire : Application des méthodes de NLP en assurance spatiale

Confidentialité : ☐ NON ☒ OUI (Durée : ☐ 1 an ☒ 2 ans)

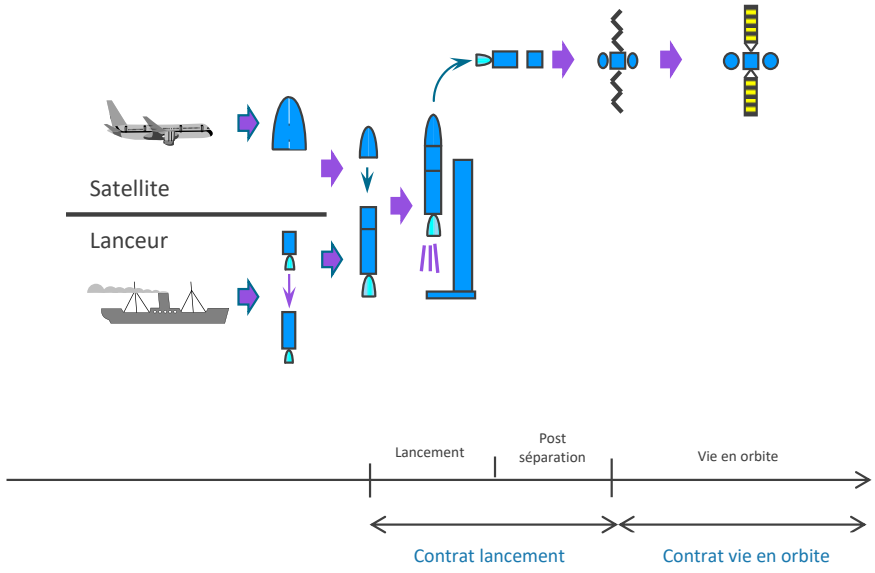
Les signataires s'engagent à respecter la confidentialité indiquée ci-dessus.

Membres présents du jury de la filière : Signature :	Entreprise : Nom : SCOR SE Signature :
	Directeur de mémoire en entreprise Nom : Hicham DAHER Signature :
Membres présents du jury de l'Institut des Actuaires : Signature :	Invité : Nom : Signature :
	Autorisation de publication et de mise en ligne sur un site de diffusion de documents actuariels (après expiration de l'éventuel délai de confidentialité) Signature du responsable entreprise : Signature du candidat :

Note de synthèse

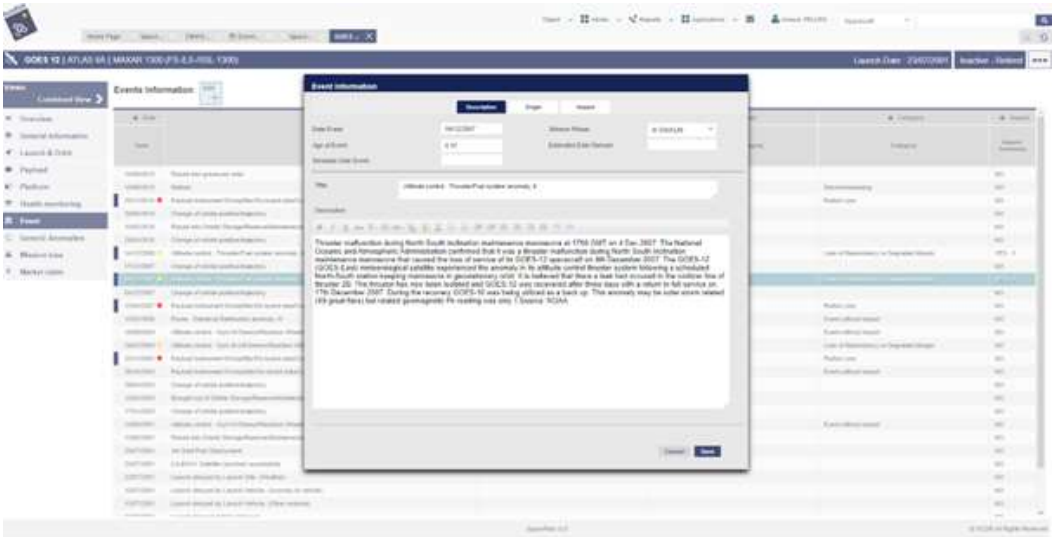
La gestion et l'analyse des risques sont des éléments fondamentaux des métiers de l'assurance spatiale. En effet, les satellites sont des engins technologiques hautement sensibles et complexes qui risquent d'être endommagés à chaque étape de leur vie. Les échecs sont fréquents lors du lancement du satellite et de sa mise en orbite. Les cas typiques d'échecs au lancement sont l'explosion du véhicule ou l'insertion du satellite sur une trajectoire orbitale incorrecte. Lors de la phase d'exploitation, des dommages mineurs peuvent entraîner une perte totale et l'échec de l'ensemble du projet.

Pour protéger leurs investissements, un grand nombre des opérateurs commerciaux des satellites de communication font appel à des organismes d'assurance couvrant le risque spatial. Le transfert des risques sur le marché de l'assurance spatiale se fait par le biais de deux grands types de contrats : le « contrat lancement » qui couvre les pertes subies lors de la phase de lancement et de post séparation et le « contrat vie en orbite » couvrant les dommages et dysfonctionnements du satellite survenus lors de la phase d'exploitation.



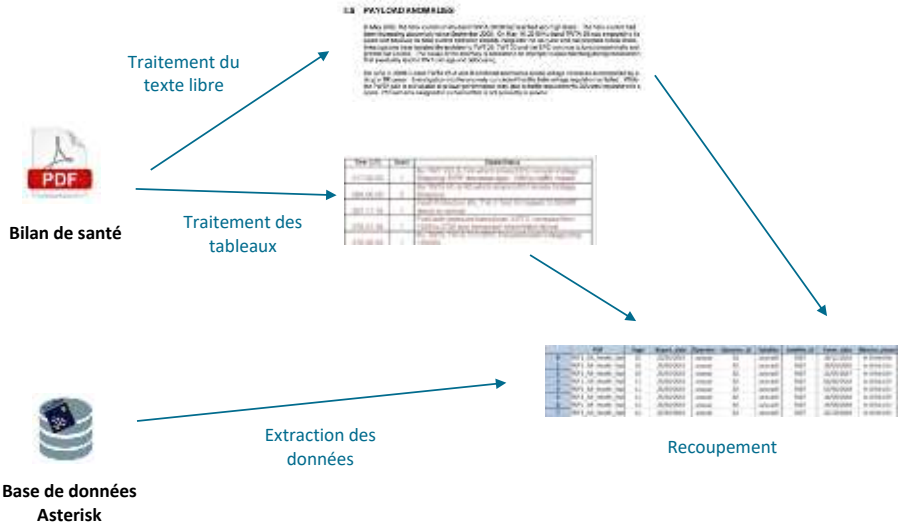
Les différents contrats du risque spatial chez Scor
Source : Scor

Chaque année, les opérateurs des satellites de communication assurés chez Scor fournissent des bilans de santé faisant notamment état des anomalies survenues dans leur flotte de satellites. Ces bilans de santé sont transmis au format PDF et détaillent les événements : la date, le satellite touché, le sous-système touché, … Autant de caractéristiques qu'il est pertinent pour un souscripteur de récupérer et de stocker dans une base de données afin de produire des études et comparatifs sur lesquels il s'appuiera dans son travail. Scor possède une telle base de données nommée Asterisk. Cette base trace toutes les anomalies déjà rencontrées et les détails les concernant. L'outil créé dans le cadre de ce mémoire visera à la remplir automatiquement grâce aux bilans de santé.



Exemple d'anomalie dans la base de données Asterisk
Source : Scor

Pour remplir automatiquement la base de données Asterisk, l'outil extrait d'abord les informations du bilan de santé considéré. Il est important lors de cette étape de tenir compte de la structure des rapports qui diffèrent selon les opérateurs. Les informations concernant les anomalies sont présentées sous la forme de textes libres ou de tableaux dans le PDF. Deux démarches différentes sont adoptées : la librairie python Tabula est utilisée pour l'extraction des tableaux tandis que la librairie PDFMiner est utilisée pour l'extraction du texte libre. Il est primordial lors de cette étape de n'extraire dans le PDF que les informations concernant les anomalies car certains tableaux et textes libres évoquent des informations générales sur les satellites. Après extraction de l'information contenue dans le PDF, un enrichissement est réalisé via la base de données Asterisk comme l'identifiant Scor du satellite touché par l'anomalie, l'identifiant de l'opérateur etc. Dans le tableau de sortie de l'outil, chaque ligne correspond à une anomalie et présente le satellite touché, la date de l'évènement, ses différentes caractéristiques etc.



Processus d'extraction et de recoupement

Le second objectif de ce mémoire est d'enrichir l'apport de l'outil créé par le biais d'un modèle de machine learning. En effet, certaines anomalies extraites ont peu d'impact sur la performance du satellite tandis que d'autres peuvent provoquer sa perte totale. Les anomalies peuvent ainsi être classées en quatre catégories de sévérité différentes :

- Pertes totales : Les pertes totales correspondent aux anomalies ayant provoqué une destruction physique du satellite ou une incapacité totale à remplir la mission
- Pertes partielles : Elles concernent les anomalies qui ont provoqué une baisse de performance du satellite, impactant la mission initialement prévue et diminuant sa durée de vie.
- Pertes de redondance ou marge dégradée :
  Les satellites incorporent généralement des redondances dans leurs sous-systèmes ou équipements (par exemple redondance des ordinateurs de bord). Ainsi, il arrive que la défaillance d'un équipement n'affecte pas la performance du satellite si celui-ci est redondant.
  De même, des marges de conception permettent aux satellites d'anticiper certaines défaillances techniques. Par exemple, une quantité d'ergol plus importante que nécessaire permet de prévenir le risque de fuite.
- Évènements sans impact :
  Ils concernent les évènements n'ayant pas entraîné de perte de performance du satellite. Aucun équipement n'a été touché et les marges de conception n'ont pas diminué.

A l'aide des descriptions des anomalies extraites par l'outil, le modèle sera capable de prédire la sévérité associée. Il s'agit d'un problème de traitement du langage naturel NLP (Natural Language Processing ou Traitement du langage naturel en français). Le NLP existe depuis plus de 50 ans et trouve ses racines dans le domaine de la linguistique. Il dispose d'une variété d'applications réelles dans un certain nombre de domaines, y compris la recherche médicale, les moteurs de recherche et l'informatique décisionnelle. Que la langue soit parlée ou écrite, le traitement du langage naturel utilise l'intelligence artificielle et le machine learning pour traiter des données du monde réel et leur donner un sens d'une manière qu'un ordinateur peut comprendre.

La base utilisée pour entraîner le modèle de classification est construite à la main par les souscripteurs du risque spatial, qui ont relevé dans les bilans de santé de nombreuses phrases évoquant une anomalie en précisant pour chacune la sévérité de l'évènement. Les données contiennent 3 194 lignes : chacune d'entre elles contient une description d'anomalie et la sévérité correspondante.

La première étape de ce projet de NLP consiste à prétraiter les données textuelles. L'objectif du prétraitement des données est tout d'abord de réduire au maximum la taille du vocabulaire utilisé. Les composantes non pertinentes du texte (ou stop words) telles que les prépositions ou déterminants sont retirées et une opération de stemming (ou racinisation en français) est réalisée pour récupérer la racine des mots.

Ensuite, il est nécessaire de transformer les mots constituant les phrases en données numériques pour qu'ils puissent être interprétés par les algorithmes. L'objectif est de représenter chacune des descriptions du corpus d'entraînement sous la forme d'un vecteur. On appelle cela la vectorisation. Trois méthodes de vectorisation sont testées dans ce projet : le bag of words, la vectorisation TF IDF (Term Frequency - Inverse Document Frequency) et la vectorisation Word2Vec.

Une fois les données prétraitées, un algorithme est développé pour la modélisation en sévérité. Il existe de nombreux algorithmes de traitement du langage naturel. Dans le cadre de ce mémoire, les modèles suivants sont entraînés : l'algorithme de machine à vecteurs de support (ou SVM), les algorithmes d'arbres tels que la forêt aléatoire (ou random forest), le boosting de gradient (ou gradient boosting) et le modèle d'apprentissage profond réseau de neurones.

Pour juger de l'efficacité d'un modèle de machine learning, il est nécessaire de se baser sur des métriques d'évaluation. Dans le cadre de ce mémoire, l'accuracy, la précision, la spécificité, le F1 score et l'aire sous la courbe ROC (AUC - Area under the ROC Curve) seront calculés. La première étape est de diviser la base de données afin d'avoir 75% des données pour l'entraînement du modèle et 25% pour la partie test. Les modèles sont entraînés et les hyperparamètres de chaque algorithme sont optimisés par validation croisée. Les modèles sont ensuite appliquées sur l'ensemble de données test et les métriques de performance sont calculées. Les résultats obtenus sont les suivants :

	SVM	**Random Forest**	**Gradient Boosting**	**Réseaux de neurones**
Bag of words	0,44	0,45	0,41	0,48
TF IDF	0,72	0,71	0,70	0,66
Word2vec	0,24	0,56	0,53	0,55

F1 score des modèles en fonction de la vectorisation utilisée

La méthode de vectorisation ayant obtenu le F1 score le plus élevé est sélectionnée. Il s'agit ici de la vectorisation TF IDF.

Ensuite, la performance de chacun des modèles entraînés sur la base vectorisée en TF IDF est évaluée avec des métriques supplémentaires : l'accuracy et l'AUC.

	SVM	**Random Forest**	**Gradient Boosting**	**Réseaux de neurones**
Accuracy	0,74	0,72	0,71	0,68
F1 score	0,72	0,71	0,70	0,66
AUC	0,92	0,91	0,90	0,89

Métriques de performance des modèles

Les résultats des quatre algorithmes sont relativement proches. L'accuracy et le F1 score se situent autour de 0,7 et l'AUC autour de 90%. Le réseau de neurones s'avère être le modèle le moins performant. Les résultats sont en revanche plus élevés pour le modèle SVM, qui est donc sélectionné pour la suite des analyses.

Dans le tableau ci-dessous sont présentés les résultats détaillés de la métrique F1 score pour la modèle SVM sélectionné, avec les valeurs de précision et de rappel.

Étiquette	Précision	Rappel	F1 score	Nombre d'observations
Évènement sans impact	0,61	0,62	0,61	106
Perte de redondance ou marge dégradée	0,66	0,70	0,68	175
Perte partielle	0,78	0,75	0,77	273
Perte totale	0,82	0,81	0,82	252
Moyenne	**0,72**	**0,72**	**0,72**	**806**

Métriques de performance pour le modèle SVM sélectionné (sur la base de données test)

Les résultats du modèle peuvent être remis en question, notamment par le faible nombre de données de la base d'entraînement (seulement 3 194 données). Un des problèmes majeurs de la classification textuelle est l'acquisition des données étiquetées afin d'entraîner les modèles. Il existe cependant une autre méthode d'apprentissage en machine learning à mi-chemin entre le l'apprentissage supervisé et non supervisé permettant de faire face à ce type de problème : la méthode semi-supervisée. Elle combine des exemples étiquetés et non étiquetés pour élargir l'ensemble de données disponibles pour la construction des modèles. Le modèle SVM sera donc entraîné une seconde fois avec 2 403 données non étiquetées supplémentaires. Les résultats obtenus sur les données test sont les suivants :

Étiquette	Précision	Rappel	F1 score	Nombre d'observations
Évènement sans impact	0.65	0.67	0.66	106
Perte de redondance ou marge dégradée	0.71	0.72	0.71	175
Perte partielle	0.78	0.73	0.75	273
Perte totale	0.83	0.81	0.82	252
Moyenne	**0.74**	**0.73**	**0.74**	**806**

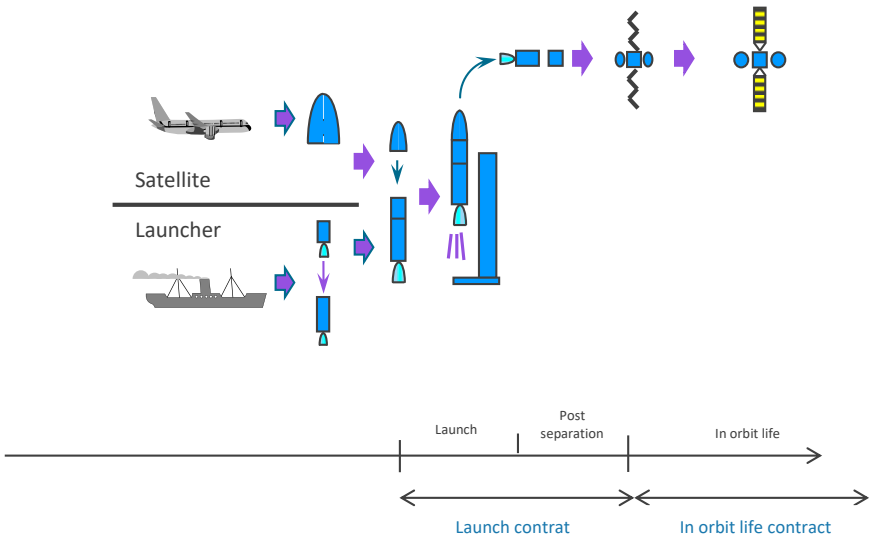
Métriques de performance obtenues avec l'apprentissage semi-supervisé

En comparant ce tableau avec celui obtenu précédemment, nous remarquons que le F1 score du modèle s'est légèrement amélioré. Il passe de 0,72 à 0,74.

Executive summary

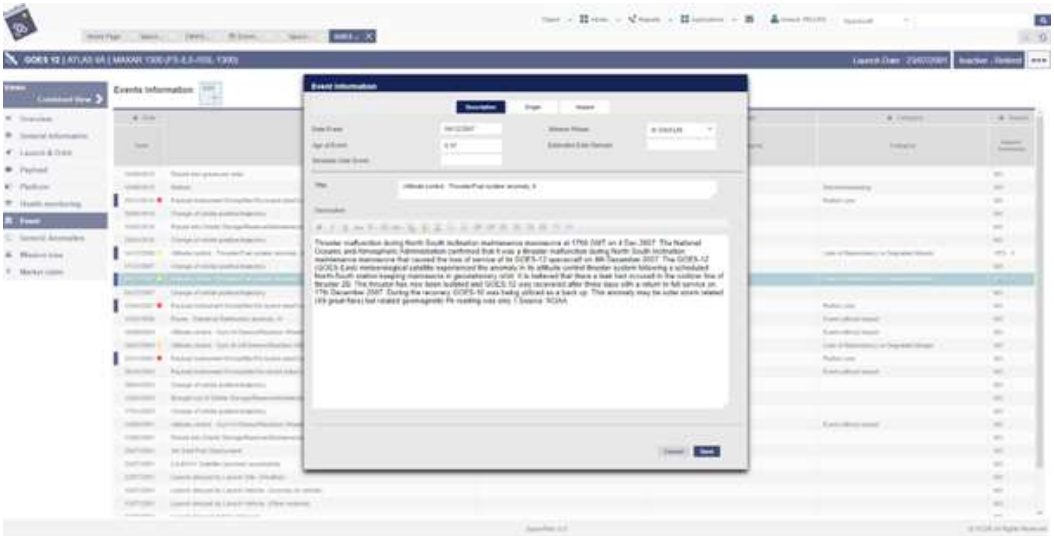
Risk management and analysis are fundamental elements of space insurance businesses. Indeed, satellites are highly sensitive and complex technological devices and risk being damaged at each stage of their life. Failures are common when launching the satellite and putting it into orbit. Typical cases of launch failures are the explosion of the vehicle or the insertion of the satellite on an incorrect orbital trajectory. During the operation phase, minor damage can lead to total loss and failure of the entire project.

To protect their investments, a large number of commercial operators of communication satellites use insurance organizations to cover space risk. The transfer of risks on the space insurance market is done through two main types of contracts : the "launch contract" which covers the losses incurred during the launch and post-separation phase and the "in orbit life contrat" covering damage and malfunctions of the satellite that occurred during the operational phase.



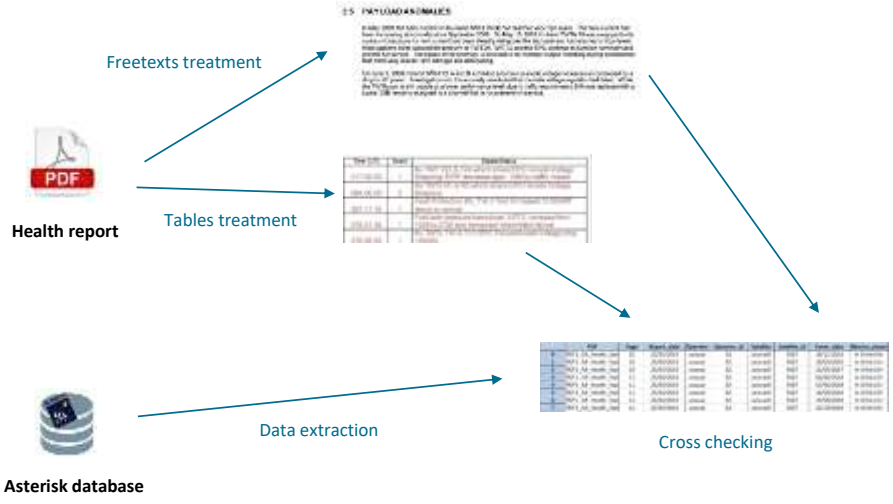
The different space risk contracts at Scor
Source : Scor

Each year, the operators of communication satellites insured by Scor provide health reports, in particular reporting on anomalies that have occurred in their fleet of satellites. These health reports are transmitted in PDF format and detail the events : the date, the affected satellite, the affected subsystem, etc. As many characteristics as it is relevant for a subscriber to retrieve and store in a database in order to produce studies and comparisons on which it will rely in its work. Scor has such a database named Asterisk. This database traces all the anomalies already encountered and the details concerning them. The tool created as part of this thesis will aim to fill it automatically thanks to health reports.



Example of an anomaly in the Asterisk database
Source : Scor

To automatically populate the Asterisk database, the tool first extracts information from the health check in question. It is important during this stage to take into account the structure of the reports which differ according to the operators. Anomaly information is presented in the form of free text or tables in the PDF. Two different approaches are adopted : the python library Tabula is used for the extraction of tables while the PDFMiner library is used for the extraction of free text. In addition, it is essential during this stage to extract only the information concerning the anomalies from the PDF because certain tables and free texts evoke general information on the satellites. After extracting the information contained in the PDF, an enrichment is carried out via the Asterisk database such as the Scor identifier of the satellite affected by the anomaly, the identifier of the operator, etc. In the tool's output table, each line corresponds to an anomaly and presents the affected satellite, the date of the event, its different characteristics, etc.



Extraction process and cross checking

The second objective of this thesis is to enrich the contribution of the tool created through a machine learning model. Indeed, some anomalies extracted have little impact on the performance of the satellite while others can lead to its total loss. Anomalies can thus be classified into four different categories of severity :

- Total loss : Total losses correspond to anomalies that caused the physical destruction of the satellite or a total inability to complete the mission.
- Partial loss : These relate to anomalies that have caused a drop in the performance of the satellite, impacting the mission initially planned and reducing its lifespan.
- Loss of redundancy or degraded margin :
  Satellites generally incorporate redundancies in their subsystems or equipment (e.g., redundancy of on-board computers). Thus it happens that the failure of a piece of equipment does not affect the performance of the satellite if it is redundant.
  Similarly, design margins allow satellites to anticipate certain technical failures. For example, a larger quantity of propellant than necessary prevents the risk of leakage.
- Event without impact : They concern events that have not led to a loss of satellite performance. No equipment was affected and design margins did not decrease.

Using the descriptions of the anomalies extracted by the tool, the model will be able to predict the class of the associated severity. This is a Natural Language Processing (NLP) problem. The NLP has been around for over 50 years and has its roots in the field of linguistics. It has a variety of real-world applications in a number of fields, including medical research, search engines, and business intelligence. Whether language is spoken or written, natural language processing uses artificial intelligence and machine learning to process real-world data and make sense of it in a way that a computer can understand.

The database used to train the classification model is built by hand by the spatial risk subscribers, who noted in the health checks many sentences evoking an anomaly, specifying for each the severity of the event. The data contains 3 194 lines : each of them contains an anomaly description and the corresponding severity.

The main stages of an NLP project are common to a classic machine learning project. The first is to pre-process the textual data. The first objective of data pre-processing is to minimize the size of the vocabulary used. Irrelevant components of the text (or stop words) such as prepositions or determiners are removed and a stemming operation is performed to recover the root of the words.

Then, it is necessary to transform the words constituting the sentences into numerical data so that they can be interpreted by the algorithms. The objective is to represent each of the descriptions of the training corpus in the form of a vector. This is called vectorization. Three vectorization methods are tested in this project : bag of words, TF IDF vectorization (Term Frequency - Inverse Document Frequency) and Word2Vec vectorization.

Once the data has been pre-processed, an algorithm is developed for severity modelling. There are many natural language processing algorithms. As part of this thesis, the following models are trained: the support vector machine algorithm (SVM), tree algorithms such as random forest, gradient boosting and the deep learning neural network model.

To judge the effectiveness of a machine learning model, it is necessary to rely on evaluation metrics. As part of this thesis, the accuracy, precision, specificity, F1 score and area under the ROC curve (AUC - Area under the ROC Curve) will be calculated. The first step is to divide the database in order to

have 75% of the data for the training of the model and 25% for the test part. The models are trained and the hyperparameters of each algorithm are optimized by cross-validation. The models are then applied to the test dataset and performance metrics are calculated. The results obtained are as follows :

	SVM	Random Forest	Gradient Boosting	Neural network
Bag of words	0,44	0,45	0,41	0,48
TF IDF	0,72	0,71	0,70	0,66
Word2vec	0,24	0,56	0,53	0,55

F1 scores models based on vectorization used

The vectorization method with the highest F1 score is selected. This is the TF IDF vectorization.

Then, the performance of each of the models trained on the vectorized TF IDF base is evaluated with additional metrics : accuracy and AUC.

	SVM	Random Forest	Gradient Boosting	Neural network
Accuracy	0,74	0,72	0,71	0,68
F1 score	0,72	0,71	0,70	0,66
AUC	0,92	0,91	0,90	0,89

Model performance metrics

The results of the four algorithms are relatively close. The accuracy and the F1 score are around 0.7 and the AUC is close to 90%. The neural network turns out to be the least efficient model. On the other hand, the results are higher for the SVM model, which is therefore selected for the rest of the analyses.

In the table below are presented the detailed results of the F1 score metric for the selected SVM model, with the precision and recall values.

Label	Precision	Recall	F1 score	Number of observations
Total loss	0,61	0,62	0,61	106
Partial loss	0,66	0,70	0,68	175
Loss of redundancy or degraded margin	0,78	0,75	0,77	273
Event without impact	0,82	0,81	0,82	252
Mean	**0,72**	**0,72**	**0,72**	**806**

Performance metrics for selected SVM model (based on test database)

The results of the model can however be called into question, in particular by the low amount of data from the training base (only 3 194 data). One of the major problems of textual classification is the

acquisition of labelled data in order to train the models. However, there is another machine learning method halfway between supervised and unsupervised learning to deal with this type of problem : the semi-supervised method. It combines labelled and unlabelled examples to expand the data set available for model building. The SVM model will therefore be trained a second time with 2403 additional unlabelled data. The results obtained on the test data are as follows :

Label	Precision	Rappel	F1 score	Number of observations
Total loss	0.65	0.67	0.66	106
Partial loss	0.71	0.72	0.71	175
Loss of redundancy or degraded margin	0.78	0.73	0.75	273
Event without impact	0.83	0.81	0.82	252
Mean	**0.74**	**0.73**	**0.74**	**806**

Performance obtained with semi-supervised learning

Comparing this table with the one obtained previously, we notice that the F1 score of the model has slightly improved. It goes from 0.72 to 0.74.

Remerciements

Je souhaite tout d'abord remercier Hicham DAHER, manager en data science chez Scor pour m'avoir suivi durant mon mémoire mais également tout au long de mon année d'alternance. Il a su faire preuve de disponibilité et de bienveillance à mon égard. Je suis persuadée que ses conseils avisés me serviront encore pour très longtemps.

Je remercie également Arnaud PELLEN, souscripteur des risques spatiaux chez Scor, pour l'ensemble des connaissances qu'il m'a apportées dans le domaine spatial et pour la relecture de mon mémoire.

Plus généralement, je remercie l'ensemble de mes collègues de l'équipe FIT (Fonctional Architecture, Innovation and Transversal Services) pour leur accueil chaleureux et leur accompagnement.

Je souhaite adresser mes remerciements à mon tuteur académique Olivier LOPEZ pour ses conseils et ses indications ainsi qu'à l'ensemble de l'équipe pédagogique de l'ISUP pour la formation qui m'a été donnée pendant trois années.

Enfin, j'ai une pensée toute particulière pour l'ensemble de mes proches et ma famille, qui m'ont accompagné, aidé, soutenu et encouragé tout au long de la réalisation de ce mémoire.

Table des matières

Introduction

Depuis le succès du premier satellite en orbite Spoutnik en 1957, les lancements des satellites n'ont cessés de se développer et sont aujourd'hui devenus courants. Alors que les premiers satellites en orbite autour de la Terre appartenaient et étaient financés par le gouvernement, les satellites commerciaux privés se sont peu à peu développés et sont aujourd'hui devenus indispensables dans divers domaines tels que la communication, la météorologie ou encore le domaine de la défense.

Le coût de construction et de lancement d'un satellite est colossal et peut généralement atteindre plusieurs centaines de millions de dollars. Pourtant, environ un satellite sur dix en moyenne est affecté par une perte de capacité (partielle ou totale) en phase lancement ou lors de sa première année de vie. De même, des pannes ou anomalies peuvent survenir lors de sa phase d'exploitation. Pour protéger leurs investissements, les acteurs du secteur spatial transfèrent les risques de pertes sur le marché de l'assurance. Actuellement, une quarantaine d'entités d'assurance participent directement à cette activité dont l'entreprise Scor qui propose, en plus de ses activités de réassurance, des services d'assurance directe dans le domaine des grands risques d'entreprise, et notamment des risques spatiaux.

La mise en place d'un contrat d'assurance spatiale, par les enjeux économiques qu'elle implique et le risque élevé de pertes financières, s'avère complexe. Une étude préalable des risques est nécessaire pour la souscription et la tarification des contrats. L'entreprise Scor dispose au sein de la branche Specialty Insurance (rattachée à la division SCOR P&C), d'une ligne de business Espace chargée d'étudier la situation de chaque satellite afin de déterminer les termes et conditions de la couverture (exposition de l'assureur, termes de la police et conditions tarifaires). Une grande partie des clients assurés dans l'entreprise sont des opérateurs commerciaux qui souhaitent couvrir les pertes des satellites de communication. Ces opérateurs envoient régulièrement des bilans de santé à l'entreprise faisant état des anomalies survenues dans leur flotte de satellites en orbite. Ils indiquent notamment la date de l'anomalie, le satellite touché, le sous-système touché, la gravité de l'anomalie, etc. Autant de caractéristiques qu'il est pertinent pour un souscripteur de récupérer et de stocker dans une base de données afin de produire des études et comparatifs sur lesquels il s'appuiera dans son travail.

L'exploitation manuelle des données textuelles et des informations contenues dans les bilans de santé demande un temps important. Les technologies actuelles permettent toutefois d'étudier ce type de données et de fournir des informations sur ces dernières en moins de temps qu'il n'en faudrait à un souscripteur humain. Ces méthodes issues du domaine de l'intelligence artificielle permettent de tirer parti des rapports textuels. Elles sont regroupées au sein d'une discipline nommée NLP (Natural Language Processing ou Traitement du langage naturel en français). Les performances des algorithmes permettent d'extraire et d'analyser les informations à partir des textes que l'être humain n'aurait été en mesure de fournir dans des délais aussi restreints.

L'extraction d'informations présentes dans les bilans de santé des satellites fera l'objet de la première partie de ce mémoire. L'objectif sera de créer un outil prenant en entrée un bilan de santé sous format PDF puis extrayant toutes les informations concernant les anomalies des satellites. Le fichier de sortie sera un fichier csv avec toutes les anomalies résumées.

La seconde partie de ce mémoire abordera la création d'un modèle visant à prédire la sévérité de chacun des évènements extraits par l'outil. Pour cette modélisation, l'utilisation du Natural Language Processing (NLP) s'avérera nécessaire. L'objectif sera d'effectuer un prétraitement des données

textuelles afin qu'elles soient lisibles par les algorithmes de classification, puis de déterminer les modèles pertinents à utiliser pour cette prédiction en sévérité. Enfin, une comparaison des différentes méthodes sera effectuée pour déterminer celle qui sera la plus adaptée.

1 Le marché de l'assurance spatiale

L'objectif de ce premier chapitre est de présenter les principales caractéristiques du marché de l'assurance spatiale. Une première partie s'attachera à la naissance de l'industrie spatiale, à ses différentes fonctions et aux acteurs du marché. Puis, la seconde section sera consacrée à l'entité assurée : le satellite. Dans cette partie sera évoquée les fonctions d'un satellite, ses phases de vie et les risques spatiaux encourus. Enfin, la troisième et dernière partie abordera les caractéristiques de l'assurance spatiale et le besoin de développement d'outils de NLP dans ce secteur.

1.1 Le secteur spatial et segmentation du marché

Le développement de l'industrie spatiale débute à partir des années 1950, au cours de la Guerre Froide. Lors de cette période, la « course à l'espace » faisait rage. Il s'agissait d'une compétition entre les États-Unis et l'Union soviétique pour développer des capacités aérospatiales, y compris des satellites artificiels, des sondes spatiales sans pilote et des vols spatiaux habités. Ce besoin d'accéder à l'espace a été motivé par l'envie des nations de briller aux yeux de la communauté internationale mais surtout d'être en mesure d'assurer la défense et la sécurité des territoires par des moyens originaux et difficilement contrôlables par l'adversaire. Le lancement du premier satellite artificiel au monde, Spoutnik I, en octobre 1957, a déclenché une peur et une anxiété intenses parmi le public américain. En réponse à ce succès technologique soviétique, la National Aeronautics and Space Administration (NASA) a été créée le 1er octobre 1958 en tant que principale agence fédérale responsable de la recherche aérospatiale et du programme spatial civil.

Depuis les années 90, les satellites sont majoritairement utilisés pour répondre à la demande grandissante des services de télécommunications. Actuellement, le marché du spatial se décompose en trois principaux secteurs d'activité :

- Le secteur commercial
- Le secteur public/institutionnel
- Le secteur militaire

Secteur commercial : L'utilisation des technologies spatiales et des données qu'elles collectent, combinée aux technologies numériques habilitantes les plus avancées, génère une multitude d'opportunités commerciales qui incluent le développement de nouveaux produits et services. Les exemples les plus classiques sont les relais de communications mobiles, la télévision par satellite et le transport de données numériques (internet et autres réseaux). Ces services de télécommunications peuvent avoir un objectif à portée régionale, à la dimension d'un continent (ex : télévision sur la péninsule arabique), ou transnational/transcontinental (ex : transfert de données Asie-Europe). Un autre exemple de l'utilisation commerciale de l'espace est l'observation de la Terre. Les données fournies par les satellites d'observation terrestre sont utilisées dans divers secteurs économiques, notamment l'agriculture de précision, les assurances, la surveillance des ressources naturelles, l'exploration pétrolière et gazière, la météorologie etc.

Secteur public/institutionnel : Le marché civil du domaine spatial concerne les opérations satellitaires jugées d'intérêt pour la nation et est directement financé par les gouvernements. Le secteur public

s'appuie sur des agences spatiales nationales ou internationales comme le CNES en France (Centre National d'Études Spatiales), la NASA aux États-Unis ou encore l'ESA en Europe (Agence Spatiale Européenne). Leur rôle est de soutenir le développement technologique avec des financements multinationaux. La collaboration internationale est très développée pour réduire les coûts individuels et éviter la duplication des efforts. Les principales opérations satellitaires sur le marché civil concernent les missions de démonstration technologique, les missions ayant un intérêt scientifique, météorologique ou encore les missions d'observation de la Terre.

Secteur militaire : Les premières explorations spatiales ont eu lieu au cours de la Guerre Froide par les États-Unis et l'Union soviétique et étaient, en partie, motivées par des considérations militaires. C'est lors de cette période que furent lancés les premiers satellites de reconnaissance, dont l'objectif est de collecter des informations sur les positions militaires d'autres pays grâce à un système optique ou radar. Aujourd'hui, de nombreux officiers militaires portent des ordinateurs portables spécialement modifiés qui s'appuient sur des données guidées par satellite pour assurer leurs positions, celles de leurs alliés et de leurs cibles.

Dans le cadre de l'utilisation commerciale de l'espace, de nombreux acteurs du service, de l'industrie manufacturière et du secteur financier sont impliqués. Le schéma ci-dessous, issu de la thèse [1] donne un aperçu des principaux acteurs et de leurs relations :

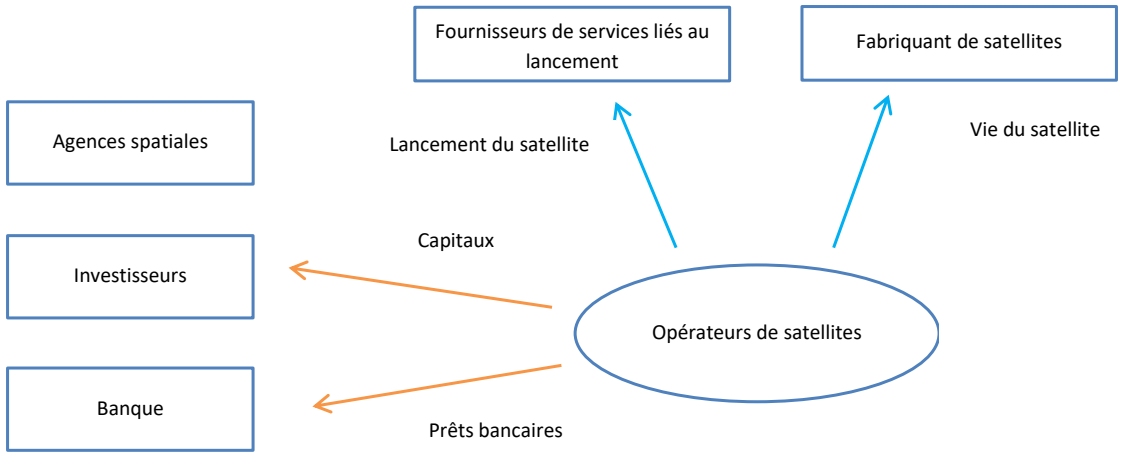


Figure 1.1 - Schéma présentant le fonctionnement de l'industrie spatiale

Les opérateurs :

L'opérateur achète et exploite une infrastructure satellitaire pour fournir un service de télécommunications ou des images satellites. Il sous-traite la fabrication des satellites et leur lancement à des entreprises privées ou publiques et fait partie intégrante de l'économie spatiale. Pour cela, les banques et les investisseurs mettent à sa disposition des capitaux. Il est bien évident que les activités d'un opérateur dépendent du bon fonctionnement des satellites engagés. Les services de télécommunications ou d'imagerie fournis par les satellites permettent à l'opérateur de générer du revenu. Cependant, l'échec du lancement d'un satellite ou l'apparition d'une faille de

fonctionnement peuvent provoquer une perte importante de revenus. Il est donc nécessaire pour les opérateurs de protéger leurs investissements.

Les constructeurs de satellites :

Les constructeurs fabriquent et vendent les satellites aux opérateurs. Lorsque le satellite est livré en orbite, le fabriquant doit supporter des risques spatiaux. Cependant, dans la majorité des cas, la livraison est faite au sol et c'est alors à l'opérateur de supporter le risque associé.

1.2 L'entité assurée : le satellite

1.2.1 Définition d'un satellite

Un satellite artificiel fait référence à un objet gravitant autour de la Terre ou d'un autre corps dans l'espace. Des milliers de satellites artificiels gravitent autour de la Terre.

Ils fournissent principalement des services de communication et d'imagerie :

- Les satellites d'observation visent à observer la Terre et collecter des informations sur ses systèmes physiques, chimiques et biologiques via des dispositifs d'imagerie en combinaison avec des capteurs non spatiaux. L'observation de la Terre permet de surveiller et d'évaluer l'état et les changements de l'environnement naturel et du paysage humain, qu'ils soient terrestres, maritimes ou aériens.

- Le satellite de communication est placé sur l'orbite terrestre afin d'établir des liaisons de communication entre différents points de la Terre. Les communications par satellite jouent un rôle vital dans le système mondial de télécommunications. Typiquement, un satellite de communication fonctionne lorsqu'il reçoit des données de stations terrestres sous forme d'ondes électromagnétiques. Les données sont envoyées par le biais d'antennes paraboliques et sont redirigées vers la station correspondante. Elles sont utilisées pour les services de télévision, radio, internet etc.

Les satellites évoluent selon trois orbites principales :

- L'orbite terrestre basse (LEO - Low Earth Orbit) allant jusqu'à 2 000 kilomètres d'altitude. Généralement, les satellites LEO sont des satellites d'observation ou de télécommunications dans le cadre de constellations (Starlink, OneWeb...).
- L'orbite terrestre moyenne (MEO – Medium Earth Orbit) située entre 2 000 et 35 786 km d'altitude.
- L'orbite terrestre géostationnaire/géosynchrone (GEO – Geostationary Orbit) avec une altitude d'environ 36 000 kilomètres. Les satellites en orbite géostationnaire sont majoritairement des satellites de communication et se déplacent à la même vitesse que la Terre.

1.2.2 Phases de vie d'un satellite et risques spatiaux

La vie d'un satellite est composée de trois phases différentes que sont la phase de lancement, la phase de post-séparation et la phase de vie en orbite. Dans cette dernière phase, le satellite est positionné sur son orbite et fournit ses services de communication ou d'observation prévus.

Phase de lancement :

Durant cette phase, le satellite est mis en orbite par un lanceur. Un lanceur est un engin à plusieurs motorisés qui se propulse par la combustion d'ergols liquides ou solides. Il permet de positionner les satellites sur leur orbite. Bien que la fiabilité des lanceurs se soit améliorée au fil des années, le lancement du satellite continue d'être une phase exposée aux aléas majeurs avec un risque de pertes particulièrement élevé. Les cas d'échecs au lancement typiques sont l'explosion du véhicule ou une mauvaise injection du satellite dans l'espace en raison d'une poussée insuffisante généralement liée à une interruption accidentelle de propulsion. La phase de lancement repose essentiellement sur les performances du lanceur alors que les phases de post-séparation et de vie en orbite dépendent de la performance du satellite.

Phase de post séparation (ou PS) :

Cette phase dure un an et est l'une des plus importantes d'une mission spatiale car elle positionne le satellite sur son orbite finale et teste ses différents éléments et équipements. Après la séparation du satellite par son lanceur, le personnel travaille continuellement pour contrôler les équipements et sous-systèmes et positionner le satellite sur la bonne orbite. Bien que des tests approfondis aient été effectués avant le lancement, l'environnement spatial ne peut pas être entièrement représenté au sol et des tests supplémentaires dans l'espace sont donc nécessaires. Les satellites sont des objets complexes composés de divers éléments mécaniques, chimiques, électriques, … Le dysfonctionnement d'un unique composant dans la phase de post séparation peut entraîner le satellite sur une mauvaise trajectoire ou l'empêcher d'atteindre une configuration exploitable.

Phase de vie en orbite (in orbit life phase ou IOL) :

La phase de positionnement est suivie de la phase d'exploitation, au cours de laquelle l'engin spatial fournit ses services de communication ou d'observation prévus. Dans l'état actuel de la technique, la durée de vie est généralement de 15 ans pour les satellites géostationnaires et de 5 à 10 ans pour les satellites en orbite terrestre basse et moyenne. Cependant, il n'est pas du tout inhabituel d'observer des durées de vie réelles plus longues que prévu. La fréquence des anomalies est plus faible que lors des deux phases précédentes. Les satellites sont toujours exposés à un environnement très hostile et sont soumis à un phénomène d'usure sans aucune possibilité de réparation.

Catégorie d'anomalies :

Les anomalies rencontrées durant la durée de vie d'un satellite, de sa phase de lancement à sa phase de mise en orbite, n'ont pas le même impact et peuvent être classées en quatre catégories de sévérité différentes :

- Pertes totales (Total loss) : Les pertes totales correspondent aux anomalies ayant provoqué une destruction physique du satellite ou une incapacité totale à remplir la mission.
- Pertes partielles (Partial loss) : Elles concernent les anomalies qui ont provoqué une baisse de performance du satellite, impactant la mission initialement prévue et diminuant sa durée de vie.
- Pertes de redondance ou marge dégradée (Loss of redundancy or degraded margin) :
  Les satellites incorporent généralement des redondances dans leurs sous-systèmes ou équipements (par exemple redondance des ordinateurs de bord). Ainsi, il arrive que la défaillance d'un équipement n'affecte pas la performance du satellite si celui-ci est redondant.
  De même, des marges de conception permettent aux satellites d'anticiper certaines défaillances techniques. Par exemple, une quantité d'ergol plus importante que nécessaire permet de prévenir le risque de fuite.
- Évènements sans impact (Event without impact) :
  Ils concernent les évènements n'ayant pas entraîné de perte de performance du satellite. Aucun équipement n'a été touché et les marges de conception n'ont pas diminué.

1.3 L'assurance spatiale

1.3.1 Généralités sur l'assurance spatiale

Il est difficile pour les acteurs du secteur spatial d'assumer seul les risques de défaillance que les satellites encourent. En effet, comme évoqué précédemment, ces derniers sont sujets à des risques de pertes significatives. Ces pertes peuvent impacter la performance de la mission initiale du satellite (par exemple mission de télécommunication). Pour protéger leurs investissements, la majorité des constructeurs et opérateurs fait appel à des organismes d'assurance afin de transférer ce risque sur le marché de l'assurance spatiale. Les compagnies d'assurance couvrent les risques de pertes du satellite depuis son lancement jusqu'à sa fin de vie.

Il existe plusieurs stratégies assurantielles permettant de couvrir les risques spatiaux. Certains acteurs disposant d'un portefeuille assez vaste peuvent se permettre d'internaliser l'ensemble des risques encourus. D'autres choisissent de faire appel à des assureurs couvrant le risque spatial. Comme pour tous les placements liés aux grands risques industriels avec des sommes assurées élevées, la plupart des polices d'assurance spatiale sont placées en coassurance. Il s'agit d'un système de compétition de l'ensemble des acteurs du marché, où chacun s'engage à prendre en charge une quote-part du risque qu'il décide de coassurer. La coassurance permet un partage horizontal du risque.

Les deux grands types de contrat dans le marché de l'assurance spatiale sont les suivants :

- Le contrat dommages directs
- Le contrat dommages causés aux tiers

L'ensemble des risques subits par le satellite est couvert par le contrat dommages directs, hors exclusions. Lorsque le satellite est dans l'incapacité de fournir les services prévus initialement, la garantie s'applique. On parle donc de garantie de performance. Dans ce type de contrat, l'assureur renonce à presque tout recours contre l'éventuel responsable du sinistre. Dans le secteur spatial, de nombreux acteurs collaborent entre eux et la moindre erreur peut entraîner des conséquences importantes sur la santé du satellite. Afin de limiter la responsabilité financière de chacun et d'empêcher une paralysie du secteur, la quasi-totalité du risque est transférée à l'assureur.

Le deuxième type de contrat couvre les dommages causés aux tiers. Cette assurance est obligatoire dans la plupart des pays dotés d'une industrie spatiale tels que la France.

Pour les contrats dommages directs, qui feront l'objet principal de ce mémoire, il existe deux types de police : le contrat lié au lancement et la phase de post séparation du satellite ainsi que le contrat lié à la vie en orbite du satellite.

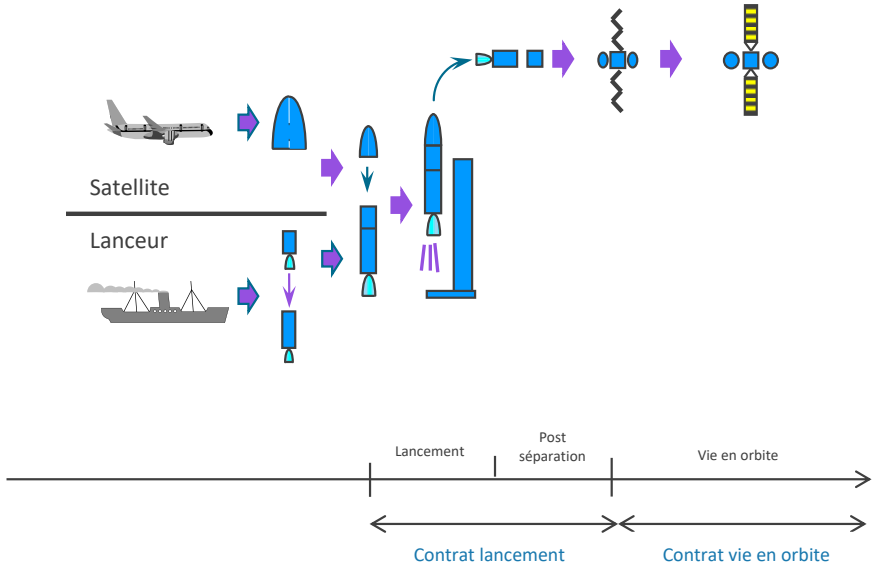


Figure 1.2 - Les différents contrats du risque spatial chez Scor
Source : Scor

Contrat lancement : Le contrat lancement couvre les pertes encourues par le satellite lors de sa phase de lancement et éventuellement de post séparation. Cette phase requiert un niveau de technologie élevé et le nombre et la sévérité des sinistres sont importants. Parmi les sinistres ayant lieu lors de cette phase, on retrouve l'explosion du lanceur, l'insertion du satellite sur une trajectoire orbitale incorrecte ou encore la panne partielle ou totale des transpondeurs lui empêchant de remplir ses fonctions. Le contrat lancement couvre le satellite sur une période de six à douze mois. La majorité des sinistres survient lors du lancement du satellite et au cours de l'insertion en orbite (respectivement 45% et 42% des sinistres). Les sinistres restants ont lieu après la phase de test.

Contrat vie en orbite : L'assurance vie en orbite couvre les dommages du satellite et les dysfonctionnements du matériel survenant lors de la phase d'exploitation. En général, des négociations de primes ont lieu entre l'assureur et l'opérateur lors du renouvellement du contrat. En effet, le montant moyen de sinistres lors de la deuxième année de mise en orbite ne représente plus qu'une fraction (de l'ordre de 10 %) de celui observé lors de la première année. Cela est dû au fait qu'une grande partie des satellites possédant des dysfonctionnements matériels subissent des dommages peu de temps après leur mise en orbite. Après la première année, le risque de panne du satellite est considérablement réduit. Néanmoins, les satellites sont exposés à un environnement très rigoureux et sont sujets à usure sans possibilité de réparation pendant de longues périodes (5 à 15 ans).

Dans la plupart des contrats spatiaux, le coût de remplacement du satellite, le prix du service de lancement ainsi que le montant de la prime d'assurance sont couverts. Les primes d'assurance représentent un montant important en raison du risque élevé de sinistralité dans le secteur spatial et du montant important des sinistres. Dans la majorité des cas, les contrats d'assurance doivent couvrir des pertes totales ou réputées totales. L'ordre de grandeur du montant assuré pour un satellite géostationnaire est entre 150 et 300 millions de dollars (M$), certains contrats pouvant dépasser les 500M$.

1.3.2 L'assurance spatiale, un secteur à caractère unique

L'assurance moderne est définie comme une relation contractuelle entre un assureur qui s'engage à couvrir un risque détenu par l'assuré en contrepartie du versement d'une prime. Elle repose sur la mutualisation des risques et est définie par des principes de base tels que :

- Un effectif important de la population assurée
- L'homogénéité de la population assurée
- La stabilité de la loi de probabilité caractérisant chaque individu

Ces trois conditions ne sont généralement pas satisfaites en assurance spatiale. Ce marché présente une activité complexe, à haut risque et possède des caractéristiques qui la rendent unique.

Assureurs spécialisés :

L'assurance des risques liés à l'espace exige une large connaissance des technologies et des risques spatiaux. La majorité des assureurs a dans ses équipes des personnes ayant une expérience dans le secteur spatial ou bien s'appuie sur des consultants externes. Les valeurs assurées peuvent atteindre plus de 500M$ et exigent la participation d'un grand nombre d'assureurs. Le marché de l'assurance spatiale est un marché international avec des assureurs situés au Royaume-Uni (Londres), en France (Paris), aux États-Unis, en Allemagne etc.

Peu de risques homogènes :

Les assureurs du risque spatial opèrent sur des échantillons peu nombreux avec des risques hétérogènes. Alors que le secteur de l'assurance s'appuie traditionnellement sur l'analyse actuarielle des données par la « mise en commun » des risques individuels ayant des caractéristiques similaires, le secteur spatial compte seulement une quarantaine de lancements assurés par an pour un stock de moins de 300 satellites assurés toujours en activité. Combiné avec une évolution continue de la technologie de cette industrie, très peu d'événements statistiques permettent d'estimer avec précision la probabilité de défaillance. La fiabilité du lanceur et du satellite, c'est-à-dire l'historique des pertes constitue donc le principal facteur déterminant de l'analyse actuarielle.

Probabilité de pertes importante et conséquences de grande envergure :

En assurance spatiale, les causes des sinistres sont nombreuses, aléatoires et régulièrement catastrophiques. Pour cette raison, les taux des primes d'assurance sont relativement élevés. L'assurance constitue le troisième coût le plus élevé dans un projet de satellite.

Bien que la navette spatiale américaine aujourd'hui à la retraite ait effectué un certain nombre de missions pour sauver des satellites défectueux, la réparation et le sauvetage en cas de dysfonctionnement sont aujourd'hui extrêmement limités. Les missions de sauvetage étant considérées comme plus coûteuses que la valeur du satellite, les opérateurs doivent s'appuyer sur la redondance des sous-systèmes et sur les ingénieurs pour trouver une solution en cas de panne.

De plus, la défaillance ou perte d'un satellite a un impact négatif non seulement sur l'opérateur, qui peut avoir besoin de trouver d'autres moyens de fournir des services tout en remplaçant un satellite

défaillant, mais aussi sur l'industrie au sens large. Lors d'un sinistre, les lancements ultérieurs du lanceur ou du satellite défaillant sont suspendus et une analyse détaillée est effectuée jusqu'à ce que les causes exactes de la perte soient déterminées. La fiabilité des lanceurs et des satellites étant un déterminant critique dans le processus de souscription, le fabricant d'un véhicule ou d'un satellite en panne et ses clients seront confrontés à des difficultés d'assurance supplémentaires. La couverture disponible peut diminuer et la prime d'assurance augmenter pour les futurs potentiels clients, rendant le satellite moins attractif pour un acheteur potentiel.

Une suspension des activités de lancement a un effet également pénalisant pour les assureurs. La prime facturée pour la couverture n'est acquise qu'au moment du lancement, ce qui affecte les revenus de l'année. La rentabilité d'un assureur des risques spatiaux se mesure par leur ratio sinistres/primes au cours d'une année civile. Un retard affecte donc directement les résultats annuels de ce dernier. En prenant en compte des sommes importantes assurées des satellites et de la moyenne des primes annuelles entre 500 millions et 800 millions de dollars, nous pouvons considérer qu'il suffit de deux échecs de lancement pour bouleverser la dynamique du marché.

Sensibilité et cycles du marché :

L'histoire montre que l'industrie de l'assurance spatiale est extrêmement sensible aux événements. A la fin des années 1980, alors que le marché du spatial sortait d'une période d'échecs, le nombre de lancements a rapidement augmenté et la capacité totale disponible a atteint un pic de 1,3 milliard de dollars. Entre les années 1998 et 2001, une combinaison de pertes en série encourus, notamment liés à des défaillances génériques sur la série de satellites BSS-601 et BSS-702, ainsi que les répercussions économiques du 11 septembre 2001, ont fortement affecté le marché de l'assurance spatiale, entraînant une diminution immédiate de la capacité et une flambée des taux de prime. Cela semble confirmer que le marché de l'assurance spatiale a un comportement cyclique. Pendant la phase "soft" du cycle, un acheteur est en mesure d'obtenir des conditions d'assurance plus favorables en raison de l'excès de capacité disponible. Cependant, lors d'une période de sinistralité accrue, les assureurs limitent leurs expositions, voire décident de quitter la ligne de business en question. Par conséquent, la capacité diminue et les primes augmentent pour assurer des revenus. Puis, les primes seront finalement réduites jusqu'à la prochaine vague de sinistres.

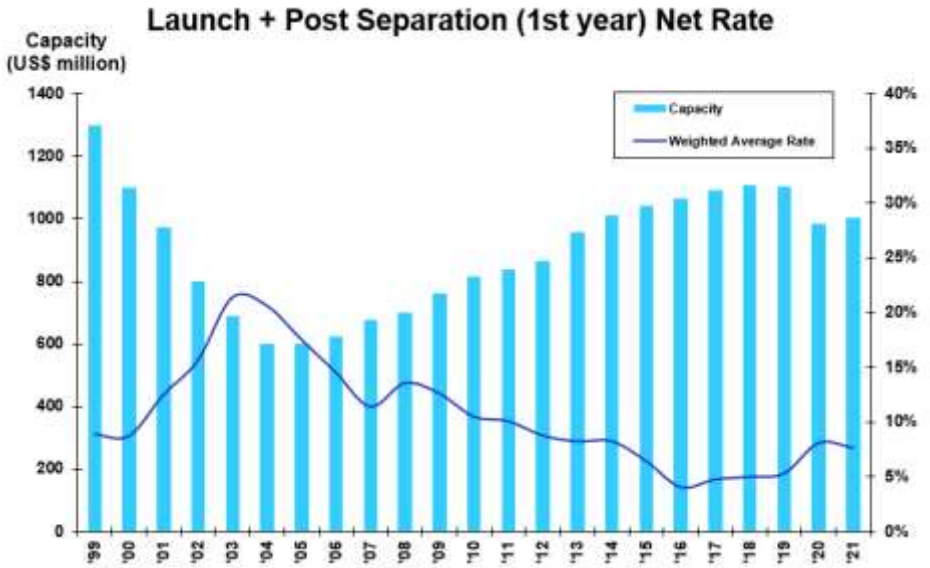


Figure 1.3 – Graphique taux de primes et capacité offerte sur le marché spatial
Source : Scor

Après une période de résultats favorables et d'augmentation régulière de capacité (les investisseurs étant attirés par le ratio primes-sinistres favorable), le marché spatial a connu une sinistralité importante en 2019, aboutissant à une contraction de la capacité disponible.

1.3.3 Le besoin de développement d'outils de NLP en assurance spatiale

Comme évoqué lors de l'introduction, l'entreprise Scor reçoit régulièrement de la part des opérateurs des bilans de santé qui font état des anomalies survenues aux satellites en orbite. Ces bilans de santé détaillent les différents événements en précisant la date, le satellite, les équipements touchés etc. Le projet de NLP détaillé dans ce mémoire vise à créer un outil capable d'extraire toutes les informations pertinentes concernant ces anomalies et de prédire leur sévérité. Ces éléments seront ensuite intégrés dans une base de données nommée Asterisk utilisée par les souscripteurs et actuaires du risque spatial. L'automatisation de l'extraction des données issues des rapports de santé serait un véritable atout pour les souscripteurs de la division Espace, la base de données Asterisk étant utilisée pour la tarification des contrats spatiaux.

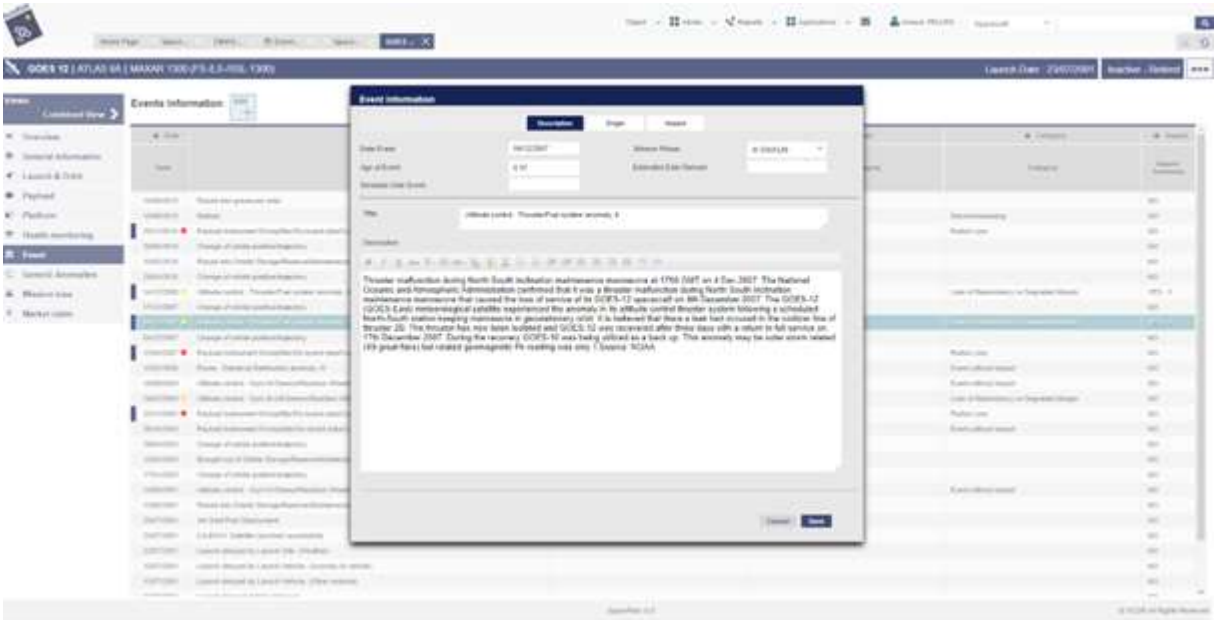


Figure 1.4 – Exemple d'anomalie dans la base de données Asterisk
Source : Scor

La partie concernant l'utilisation de la base de données Asterisk dans le processus de tarification des contrats spatiaux est située en annexe 2.

2 Création de l'outil d'analyse des bilans de santé

L'objectif de ce second chapitre est de présenter les différentes étapes de la création de l'outil d'analyse des bilans de santé des satellites. Une première partie s'attachera à la structure des documents PDF à analyser et aux informations pertinentes à extraire de ces rapports. Les deux parties suivantes présenteront la démarche d'extraction de ces informations sous forme de tableaux ou de textes libres. Enfin, la quatrième et dernière partie abordera l'enrichissement des informations extraites par des données complémentaires et la création du tableau de sortie de l'outil.

2.1 Présentation des données à analyser : les rapports de santé des satellites

L'outil prend en entrée un bilan de santé fourni par un opérateur au format PDF. Ce bilan de santé contient de nombreuses informations sur l'état des satellites de l'opérateur en question. Il comprend des tableaux listant les anomalies des différents satellites mais également des commentaires textuels décrivant plus en profondeur les évènements. Les opérateurs traités concernent les services de télécommunication. Chaque opérateur possède un certain format de rapport et le traitement des fichiers de sept opérateurs a été mis en place.

Le document ci-dessous est un extrait d'un rapport de santé d'un opérateur de satellite. On distingue les commentaires textuels sur les anomalies rencontrées en orbite ainsi que le tableau résumant l'ensemble des anomalies en bas de page.

P operator

1.1 Satellite : [redacted]

Launch date : [redacted]
Manufacturer : [redacted]

Satellite anomalies :

- S-band communications payload is subject to severe degradation of the EIRP and G/T performances, due to an S-band reflector anomaly.
- Ku-band, the 6 G-channels are subject to degradation of the G/T performance by about [redacted], due to a metallic element in the input section waveguide.
- Loss of 29 temperature indications from one TCH (Thermal Conditioning Hybrid). The correct thermal conditioning continues according to design by the effective majority vote system of temperature acquisitions, two acquisitions remaining healthy for each involved conditioning circuits, then a software patch would be performed to rely on the single remaining acquisition.

List of satellite anomalies :

Date of event	Subsystem	Anomaly description
[redacted]	Payload	Increased wheel friction
[redacted]	Eps	Short term contamination of earth sensor assembly (ESA) telescopes
[redacted]	Payload	Solar Array degradation exceeds predicts
[redacted]	Eps	Thruster 5 transient underperformance observed during LEOP orbit raising
[redacted]	Eps	Thruster orientation mechanism (TOM) initial position
[redacted]	Adcs	High accuracy pressure transducer 3, which was observed anomalous at launch site remains in failed state

Figure 2.1 – Extrait d'un rapport de l'opérateur P (tableau récapitulatif et texte libre)
Pour des raisons de confidentialité, ceci est une reproduction et non un véritable extrait d'un rapport de santé.

Le tableau récapitulatif liste ici les différentes anomalies des satellites avec leur date, le titre et le sous-système touché. Le texte libre apporte des détails supplémentaires sur ces anomalies.

Les tableaux d'anomalies des opérateurs suivent la même logique : chaque ligne des tableaux correspond à une anomalie spécifique avec des informations sur l'évènement. On retrouve la plupart du temps les colonnes suivantes : « Titre de l'anomalie », « Date de l'anomalie », « Description de l'anomalie ».

De nombreux opérateurs ne disposent pas de format normalisé et l'organisation des bilans peut changer d'une année à l'autre. Les rapports traités sont ceux des opérateurs les plus importants, à la structure pérenne et qui propose tout de même une certaine régularité dans les rapports.

On trouve également dans chaque rapport beaucoup d'informations techniques à ignorer sous forme de tableau ou de texte libre. Par exemple, les rapports de l'opérateur B débutent systématiquement par un rappel de l'ensemble des satellites en orbite, suivi d'informations techniques d'ordre général, et traite de chaque satellite un à un, avec à chaque fois des informations techniques spécifiques. Les tableaux récapitulatifs des anomalies et leur description sous forme de texte libre sont, eux, regroupés dans les dernières pages du PDF.

Name	[redacted]
Launch date	[redacted]
Launch vehicule	[redacted]
Propellant life	Last analysis show the predicted life time of … (TBC) from BOL
IOT information	The spacecraft reached its geaostationary in-orbit test slot of [redacted] degrees east after four AFM burns. Bus and payload testing began on [redacted], [redacted] and completed on [redacted], [redacted].
Orbital position	[redacted] degrees east longitude

Figure 2.2 – Extrait d'un tableau rassemblant les informations générales du satellite C (tableau non pertinent)
Pour des raisons de confidentialité, ceci est une reproduction et non un véritable extrait d'un rapport de santé.

L'outil doit extraire en premier lieu les tableaux d'anomalies, puis les textes d'intérêt présents autour des tableaux. Toutes ces informations sont ensuite mises en commun selon un format identique à tous les opérateurs.

2.2 Extraction des tableaux récapitulatifs

La récupération des tableaux se divise en deux étapes. La première étape consiste à détecter les tableaux récapitulatifs d'anomalies. Ensuite, les tableaux pertinents sont extraits et transformés en Dataframe (structure disponible dans la librairie Pandas de Python permettant de stocker des tableaux de données).

2.2.1 Détection des tableaux pertinents

Avant de procéder à la phase d'extraction, il est nécessaire de détecter les tableaux récapitulant les anomalies des satellites. Pour ce faire, les pages du PDF sont parcourues une à une afin de trouver un titre compris dans une liste des titres récurrents correspondants aux sections des tableaux d'anomalies pour l'opérateur en question. On appelle cela la détection par invariant. Le tableau de description d'anomalies ci-dessous est extrait d'un bilan de santé de l'opérateur C :

Anomaly summary

Subsystem related	Event analysis	Date of events
ACS	Happened at 06:18 on [redacted] ACS level 3 1553 bus monitor tripped and took recovery actions; 0.85 deg pitch transcient lased about 2 minutes during transition between the processors, attitude recovered with no issues.	[redacted]
DHE	Happened on [redacted] There was a few seconds of telemetry interruption from DHE-2 during the reset while the satellite was under full control without any payload traffic interruption.	[redacted]
DHE	Happened on [redacted] Several software configuration statuses changed in DHE-1 internal interface; fault protection logic tried to swap from ACE-2 to ACE-1 but it did not happen (later actions showed DHE-1 could not command to turn on ACE-1 through the internal interface)	[redacted]

Figure 2.3 – Tableau récapitulatif pour l'opérateur C
Pour des raisons de confidentialité, ceci est une reproduction et non un véritable extrait d'un rapport de santé.

Le titre « Anomaly summary » précède la plupart des tableaux décrivant les anomalies pour cet opérateur. De même, les en-têtes ont une formulation récurrente. Nous avons tout d'abord « Subsystem related », « Event analysis » puis « Date of events ».

Bien que répondant à nos besoins, la détection des tableaux par invariant a ses limites car si l'opérateur venait à changer complètement la formulation de ses titres ou en-têtes, les pages contenant un tableau pertinent ne seraient plus détectées. Dans le cadre du projet, l'utilisation principale de l'outil développé vise à intégrer à la base de données AsteRisk des rapports archivés, dont le format est de fait stable, la méthode de détection par invariant sera donc privilégiée.

2.2.2 Transformation des tableaux en Dataframe

Une fois leur emplacement identifié, les tableaux sont extraits. La librairie Python utilisée pour réaliser cette tâche est Tabula. Tabula est l'une des solutions de référence en matière d'extraction de données tabulaires, décrite comme étant la meilleure dans le domaine. En plus de permettre la conversion automatique d'une page PDF contenant un tableau en un Dataframe Pandas, cette librairie dispose de plusieurs options d'extraction rendant possible la détection et le traitement automatique de tableaux de formats différents. La méthode Lattice détecte les tableaux à partir des lignes de démarcations entre les cellules tandis que la méthode Stream étudie les variations de densités des caractères afin de simuler la structure d'un tableau.

L'image ci-dessous illustre le fonctionnement de Tabula avec la méthode Stream :

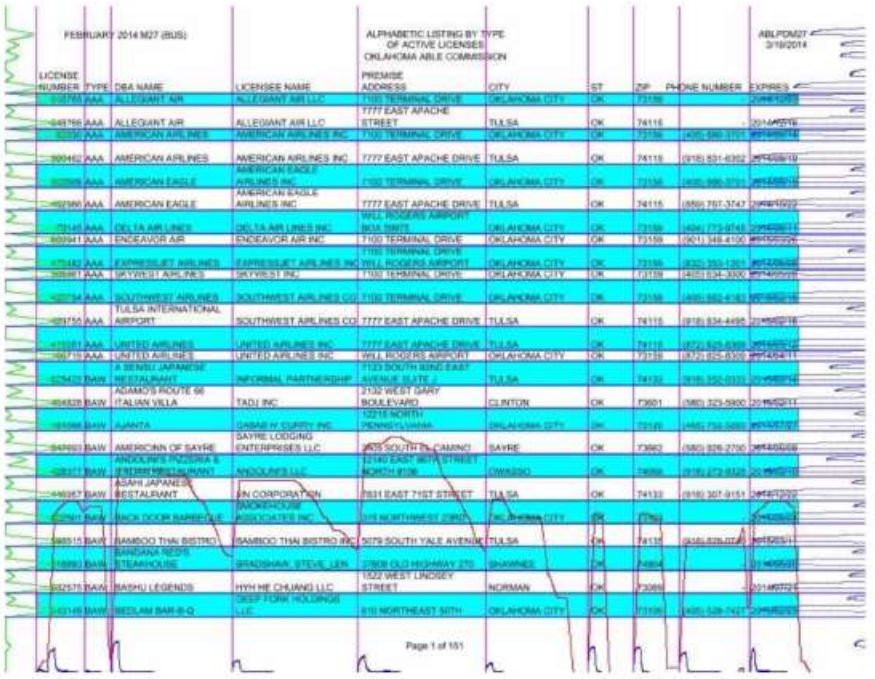


Figure 2.4 – Illustration du fonctionnement de Tabula avec la méthode Stream
Source : github.com/tabulapdf/tabula

Tabula place une ligne horizontale ou verticale à chaque fois qu'il y a une brusque variation du nombre de caractères. Or certains tableaux avec des cases un peu trop remplies et côte à côte, ou encore des lignes ne faisant pas toute la largeur du tableau, ou des colonnes ne faisant pas toute sa hauteur, empêchent une extraction correcte du contenu. Ainsi, il est parfois plus difficile pour Tabula d'extraire certains tableaux, notamment lorsqu'il ne s'agit pas de "tableau d'anomalies simple" (avec des lignes de démarcations entre les cellules).

Une fois les tableaux de chaque page extrait, il reste à regrouper les tableaux étalés sur plusieurs pages. Selon les opérateurs, la règle utilisée peut être : "si deux tableaux se succèdent dans une même partie et sont au même format, on les concatène", "si un tableau sans en-tête de colonne succède un autre tableau, on les concatène", etc.

2.3 Extraction du texte libre

2.3.1 Détection et extraction des paragraphes

Dans la partie précédente, nous avons évoqué la méthode appliquée pour récupérer toutes les informations contenues dans les tableaux récapitulatifs que nous analyserons par la suite. Il faut maintenant récupérer le texte qui se trouve autour de ces tableaux, en prenant soin de ne récupérer que des phrases évoquant une anomalie. Nous utilisons dans cette partie une librairie propre à Python et permettant facilement d'extraire du texte présent dans un PDF : la librairie PDFMiner.

Tout comme les tableaux d'anomalies, les descriptions des évènements situées en dehors des tableaux sont précédées et suivies par des titres de section propres à chaque opérateur. Le texte ci-dessous est extrait d'un PDF de l'opérateur K. Nous remarquons que la description de l'anomalie est précédée du titre « 6.3 Payload anomalies ». Ce titre est récurrent dans les rapports de santé de cet opérateur, et désigne toutes les anomalies touchant la charge utile du satellite.

6.3 Payload anomalies

In [redacted], the helix current on [redacted] had reached very high levels. The helix current had been increasing abnormally since [redacted] On [redacted] was swapped to its spare unit because its helix current had been steadily rising over the last year and had reached critical levels. Investigations have isolated the problem to [redacted], [redacted] and the EPC continue to function nominally and provide full service. The cause of the anomaly is believed to be improper output matching during construction, that enventually lead to TWT damage and defocusing.

On [redacted], [redacted] exhibited anomalous anode voitage increases accompanied by a drop in RF power. Investigation into the anomaly concluded that the helix voltage regulator had failed. While the [redacted] pair is still usable at a lower performance level, due to traffic requirements [redacted] was replaced with a spare. [redacted] remains assigned to a channel that is not presently in service.

On [redacted]. spuriously muted. Investigations identified an SET event as the most likely cause of the state change. The unit was commanded back to its nominal configuration.

Figure 2.5 – Extrait du rapport de santé de l'opérateur K
Pour des raisons de confidentialité, ceci est une reproduction et non un véritable extrait d'un rapport de santé.

Une fois les descriptions détectées puis extraites, les titres, en-têtes et pieds de page sont retirés.

2.3.2 Association de chaque paragraphe au tableau récapitulatif correspondant

L'étape suivant l'extraction du texte libre est l'association de chacune des descriptions à une anomalie figurant dans l'un des tableaux extraits. Pour ce faire, il existe différentes méthodes. La première consiste à utiliser le code spécifique aux anomalies. Pour certains opérateurs, chaque anomalie du tableau récapitulatif est désignée par un code spécifique. C'est le cas par exemple pour l'opérateur Z :

Code	Date	Event description	Status
[redacted]	[redacted]	Angular momentum growth slop increase due to TOTS deformation	Under investigation
[redacted]	[redacted]	ImpEHT-14 degradation	Under investigation

Figure 2.6 – Extrait d'un tableau d'anomalies provenant de l'opérateur Z
Pour des raisons de confidentialité, ceci est une reproduction et non un véritable extrait d'un rapport de santé.

Si l'on remarque la présence de l'un des codes du tableau récapitulatif dans un texte libre, il est facile de le rattacher à la ligne associée.

Une autre méthode générale à tous les opérateurs de satellites consiste à détecter les équipements présents dans la colonne « Descriptions d'anomalies » des tableaux extraits et dans les textes libres à partir d'une liste d'équipements récurrents des satellites. Si l'on retrouve des équipements similaires, alors le texte extrait sera associé à l'anomalie considéré et les colonnes « Texte libre » et « Equipements » seront ajoutées au tableau.

2.4 Regroupement et finalisation

2.4.1 Enrichissement avec des données complémentaires présents dans le PDF

Après l'extraction des données des tableaux récapitulatifs et des textes libres, il subsiste des informations non extraites dans le PDF. Ces informations complémentaires seront utilisées pour enrichir le tableau de sortie de l'outil d'analyse. De même, l'extraction d'éléments présents dans la case « Texte libre » des tableaux d'anomalies permettront d'enrichir notre tableau final.

Pour certains opérateurs, le tableau récapitulatif d'anomalies ne contient pas de colonne « Date de l'évènement ». Il convient donc d'extraire cette information par le biais d'une autre méthode. Il est nécessaire d'utiliser Regex, une librairie Python permettant l'extraction d'éléments à partir d'expressions régulières. La date est ainsi détectée dans le PDF à partir de l'ensemble des formats possibles (numérique ou littéral, abrégé ou non) puis extraite.

La librairie Regex permet d'extraire d'autres informations telles que la date du rapport ou bien le nom du satellite touché par l'anomalie. Les noms des satellites d'un même opérateur suivent généralement le même format et peuvent être retrouvés en première page du PDF ou bien aux alentours du tableau récapitulatif d'anomalies.

L'enrichissement du tableau de sortie de l'outil d'analyse est également réalisé par le biais de modèles de machine learning. Les textes libres extraits sont utilisés pour déterminer la sévérité des évènements. Les modèles utilisés pour cette classification ainsi que les étapes nécessaires à leur construction seront décrits en seconde partie du mémoire.

2.4.2 Enrichissement avec la base de données existante Asterisk et homogénéisation des Dataframes

L'enrichissement du tableau de sortie de l'outil se fait également grâce aux informations présentes dans la base de données Asterisk. On retrouve dans cette base des informations sur le satellite touché par l'anomalie telles que son identifiant Scor ou sa date de lancement. L'âge du satellite à la date de l'évènement est également calculé et la phase de mission est déterminée comme suit :

$$Phase\ de\ mission = \begin{cases} Post\ separation & si\ \hat{a}ge\ du\ satellite < 0.5\ an \\ In\ orbit\ life & sinon \end{cases}$$

La dernière étape est la comparaison de toutes les anomalies extraites avec celles déjà présentes dans la base de données Asterisk. Pour chaque évènement extrait du bilan de santé, on cherche ceux ayant touché le même satellite à la même date sur le même équipement dans l'historique. Si une anomalie semblable avait déjà été référencée, on l'indique grâce à l'identifiant de l'évènement dans Asterisk. Sinon, on considère que l'anomalie n'a jamais été saisie dans la base de données.

Pour chaque opérateur, la matière en sortie pour un PDF spécifique est un fichier csv qui comporte les colonnes suivantes :

	PDF	Page	Report_date	Operator	Operator_id	Satellite	Satellite_id	Event_date	Mission_phase
0	PDF1_ [redacted] _Health_Stat	10	23/04/2019	[redacted]	82	[redacted]	[redacted]	[redacted]	In Orbit Life
1	PDF1_ [redacted] _Health_Stat	10	23/04/2019		82				In Orbit Life
2	PDF1_ [redacted] _Health_Stat	10	23/04/2019		82				In Orbit Life
3	PDF1_ [redacted] _Health_Stat	11	23/04/2019		82				In Orbit Life
4	PDF1_ [redacted] _Health_Stat	11	23/04/2019		82				In Orbit Life
5	PDF1_ [redacted] _Health_Stat	11	23/04/2019		82				In Orbit Life
6	PDF1_ [redacted] _Health_Stat	12	23/04/2019		82				In Orbit Life
7	PDF1_ [redacted] _Health_Stat	12	23/04/2019		82				In Orbit Life

Tableau de sortie – partie 1

Launchdate	Age_at_event	Title	Event_description	Freetext	Similar_event_id	Equipment	Severity
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]		Earth sensor	Loss of Redundancy or
					1404	Imaging Instrument	Loss of Redundancy or
					1823	Solar array drive ass	Loss of Redundancy or
						Solar Array	Loss of Redundancy or
						Solid State Power A	Partial LossPartial Loss
					2685	Earth sensor	Loss of Redundancy or
						Thruster (chemical)	Partial LossPartial Loss
						On-board computer	Loss of Redundancy or

Tableau de sortie – partie 2

Figure 2.7 – Extraits d'un fichier csv de sortie de l'outil

Chaque ligne du tableau correspond à une anomalie extraite du bilan de santé. Les colonnes indiquent les éléments suivants :

- PDF : nom du PDF analysé
- Page : numéro de la page du PDF où a été extraite l'anomalie
- Report date : la date du rapport PDF
- Operator : le nom de l'opérateur
- Operator_id : l'identifiant de l'opérateur dans la base de données Scor
- Satellite : nom du satellite touché
- Satellite_id : l'identifiant du satellite touché dans la base de données Scor
- Event_date : date de l'anomalie
- Mission phase : phase de mission du satellite lors de l'évènement
- Launchdate : date de lancement du satellite touché
- Age_at_event : âge du satellite lors de l'évènement (en année)
- Title : titre de l'anomalie
- Event_description : description de l'anomalie
- Freetext : description de l'anomalie extraite du texte libre
- Similar_event_id : identifiant de l'anomalie similaire dans la base de données Asterisk
- Equipment : équipement du satellite touché par l'anomalie
- Severity : sévérité prédite de l'anomalie

Sur le schéma ci-dessous est représenté le processus complet d'extraction et de recoupement de l'outil :

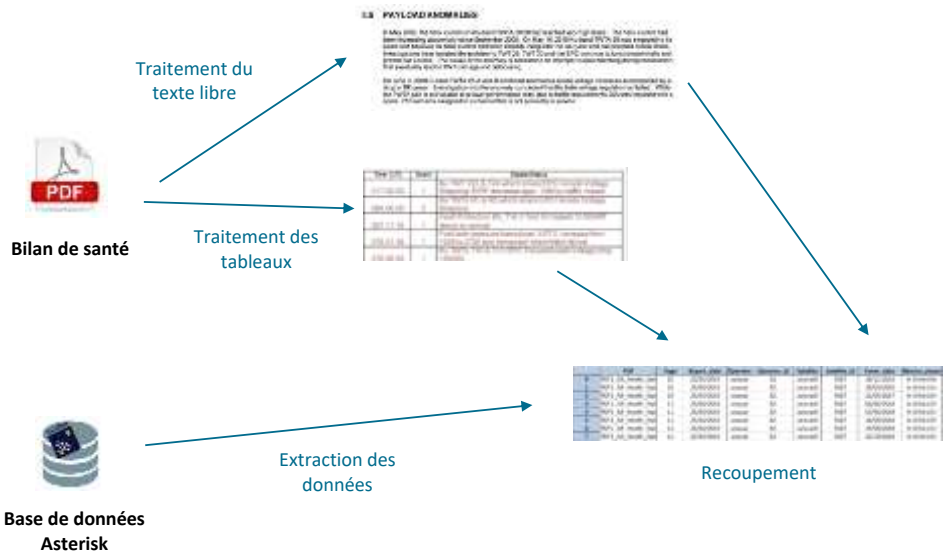


Figure 2.8 – Processus d'extraction et de recoupement

3 Classification de la sévérité des anomalies

L'objectif de cette troisième partie est de présenter la démarche de modélisation effectuée pour la classification des anomalies en sévérité. Tout d'abord seront évoquées les notions de base de machine learning et du traitement du langage naturel. Ensuite, nous décrirons la base de données utilisée pour l'entraînement des modèles de classification ainsi que les différentes étapes de prétraitement. Une quatrième partie s'attachera aux différents modèles utilisés et à leurs caractéristiques. Enfin, les résultats obtenus seront présentés et une comparaison des modèles sera effectuée.

3.1 Notion de bases en Machine Learning

Le machine learning est un ensemble de technologies d'intelligence artificielle permettant d'apprendre à partir de gros volumes de données nommées données d'apprentissage. C'est en 1959 que l'informaticien américain Arthur Samuel évoqua pour la première fois ce terme pour son programme capable de jouer aux échecs. Ce programme apprenait progressivement des différentes parties et de ses erreurs. Après avoir analysé et appris un grand nombre de parties, il finit par battre le quatrième meilleur joueur des Etats Unis.

Actuellement, il existe deux grands types de méthodes de machine learning : l'apprentissage supervisé et l'apprentissage non supervisé.

Apprentissage supervisé : La méthode d'apprentissage supervisé est la plus courante et sera utilisée dans le cadre de ce mémoire. Il s'agit de fournir au modèle une base de données d'apprentissage de la forme (X, Y) avec X les variables explicatives et Y la variable à expliquer. L'objectif du modèle est de trouver une relation entre X et Y grâce à une fonction F telle que $F(X) \approx Y$. Lorsque Y est une variable continue, il s'agit d'un modèle de régression et lorsqu'elle est définie en classe il s'agit d'un modèle de classification. L'apprentissage supervisé est une méthode qui peut être utilisée par exemple pour la prédiction du prix d'une voiture en fonction de ses caractéristiques (taille, options, etc).

Apprentissage non supervisé : Comme son nom l'indique, l'apprentissage non supervisé est une technique d'apprentissage automatique dans laquelle les modèles ne sont pas supervisés à l'aide d'un ensemble de données d'apprentissage. Au lieu de cela, les modèles eux-mêmes trouvent les modèles cachés et les informations à partir des données fournies. Les modèles sont formés à l'aide d'une base non étiquetée. Le but est de trouver la structure sous-jacente de l'ensemble de données, de les regrouper en fonction de leurs similitudes et de les représenter dans un format compressé. Cela peut être par exemple le regroupement de données représentant des clients (segmentation des clients d'un magasin par rapport à leur historique d'achats, aux produits consultés sur le site en ligne etc).

Dans le cadre de ce mémoire, nous appliquerons des méthodes de machine learning afin de manipuler, traiter et prédire des informations présentes dans la donnée textuelle. Cette discipline portant sur la compréhension du langage naturel par les machines est une branche de l'intelligence artificielle nommée Natural Language Processing (ou NLP).

3.2 Le NLP, un domaine de l'apprentissage automatique

Le NLP existe depuis plus de 50 ans et trouve ses racines dans le domaine de la linguistique. Il dispose d'une variété d'applications réelles dans un certain nombre de domaines, y compris la recherche médicale, les moteurs de recherche et l'informatique décisionnelle. Que la langue soit parlée ou écrite, le traitement du langage naturel utilise l'intelligence artificielle et le machine learning pour traiter des données textuelles du monde réel et leur donner un sens d'une manière qu'un ordinateur peut comprendre. Il existe différentes applications possibles du NLP dont :

- La reconnaissance vocale : Cette technologie permet à l'ordinateur de convertir les données d'entrée vocale en format lisible par machine. Elle est par exemple utilisée dans les moteurs de recherche où l'utilisateur peut prononcer le nom de ses exigences de recherche et obtenir le résultat souhaité plutôt que de taper la commande entière.
- Correction automatique et prédiction automatique : De nos jours, il existe de nombreux logiciels disponibles qui vérifient la grammaire et l'orthographe du texte que nous tapons et nous évitent des fautes d'orthographe et de grammaire dans nos e-mails, textes ou autres documents. C'est l'une des applications les plus utilisées du NLP. Il existe également des fonctionnalités permettant de prédire automatiquement le texte que nous avons commencé à taper. Cela fait gagner du temps à l'utilisateur et lui facilite la tâche.
- La classification de textes : La classification de textes est une technique d'apprentissage automatique qui attribue un ensemble de catégories prédéfinies à un texte. Elle est l'une des tâches fondamentales du traitement du langage naturel avec de vastes applications telles que l'analyse des sentiments, l'étiquetage des sujets, la détection des spams ou encore la détection des intentions. Dans le cadre de ce mémoire, la classification sera réalisée afin de prédire quatre types de sévérité d'anomalies.

Il existe deux étapes primordiales à l'élaboration d'un modèle de NLP : le prétraitement des données et le développement de l'algorithme :

- La première étape est le prétraitement des données. Elle consiste à préparer et à "nettoyer" les données textuelles pour que les modèles puissent les analyser.
- Une fois les données prétraitées, un algorithme est développé pour la modélisation de la sévérité. Il existe de nombreux algorithmes de traitement du langage naturel. Dans le cadre de ce mémoire, nous allons entraîner les modèles suivants : l'algorithme de machine à vecteurs de support (ou SVM), les algorithmes d'arbres tels que la forêt aléatoire (ou random forest) et le boosting de gradient (ou gradient boosting) et le modèle d'apprentissage profond réseau de neurones. Ces modèles sont applicables dans le cas d'une classification multiclasse. De plus, ils sont efficaces sur des jeux de données avec plus de variables que d'observations. Une fois les modèles construits, nous déterminerons le modèle le plus performant à l'aide de métriques d'évaluation et de comparaison.

3.3 Présentation et prétraitement des données disponibles pour la classification

3.3.1 Présentation de la base et des variables pour entraîner le modèle

La base utilisée pour l'entraînement des modèles de classification est construite à la main par les souscripteurs du risque spatial, qui ont relevé dans les bilans de santé de nombreuses phrases évoquant une anomalie en précisant pour chacune la sévérité de l'évènement.

Les données contiennent 3194 lignes : chacune d'entre elles contient une description d'anomalie et la sévérité correspondante. Comme présenté dans la partie I, la sévérité d'un évènement peut être classée selon quatre types différents :

- Perte totale
- Perte partielle
- Perte de redondance ou marge dégradée
- Évènement sans impact

Ci-dessous est présenté un extrait de la base de données au format Excel. On remarque bien à gauche la description de l'anomalie et à droite la sévérité associée :

	Description	Target
0	Sudden increase in friction torque of [redacted]	Loss of Redundancy or Degraded Margin
1	Pointing transient d[redacted] caused	Partial Loss
2	[redacted] experienced 2 unexpected return to [redacted]	Loss of Redundancy or Degraded Margin
3	Date of Occurrence estimated with information provided by [redacted]. Has e	Loss of Redundancy or Degraded Margin
5	Accurate date unknown, but estimated date as per [redacted] prelaunch Goo	Loss of Redundancy or Degraded Margin
6	Friction starts to increase on [redacted]	Event without impact
9	Friction increase starting after launch	Event without impact
11	An increase in dry fri[redacted] a potential problem with [redacted] v	Loss of Redundancy or Degraded Margin
13	[redacted] N[redacted] on [redacted] reported excessive fric	Loss of Redundancy or Degraded Margin
14	[redacted] On 1[redacted] spacecraft experienced an anc	Loss of Redundancy or Degraded Margin
20	On [redacted], just prior [redacted] has observed triggering of t	Event without impact
21	The momentum wheel [redacted] anormally powered off, resulting in the	Event without impact
22	[redacted] off on [redacted]. The unit has been reactivated and is working	Event without impact
23	In [redacted] the friction on wheel [redacted] that appeared in [redacted] and was cor	Event without impact
25	On [redacted] spontaneously shut off. As a result earth lock was lost	Event without impact
26	On [redacted] spontaneously shut off. The wheel was comman	Event without impact

Figure 3.1 – Extrait de la base de données d'entraînement

Sur le schéma ci-dessous est présenté le nombre de descriptions d'anomalies de la base de données en fonction du label associé :

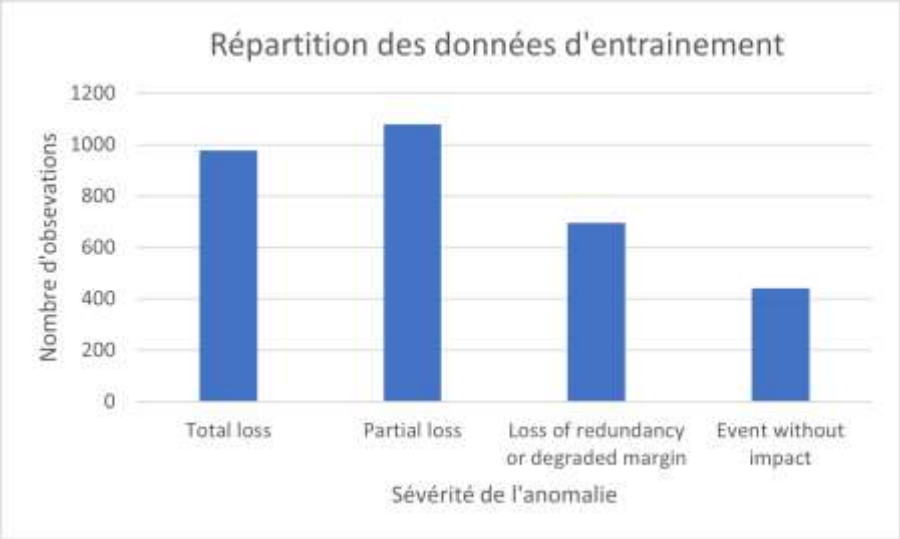
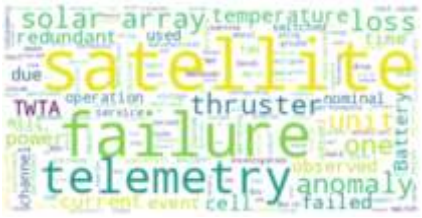


Figure 3.2 – Répartition des données d'entraînement

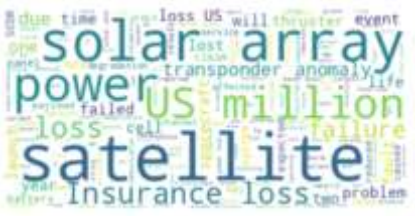
On s'aperçoit que la base contient d'avantage d'anomalies classées comme « Perte partielle » et « Perte totale » (environ 1 000 par catégorie) que d'évènements « Perte de redondance ou marge dégradée » et sans impact. Nous allons devoir tenir compte du fait que l'ensemble de données est déséquilibré lors de l'entraînement de nos modèles en utilisant par exemple des métriques d'évaluation de performance adaptées.

Les mots présents dans les descriptions d'anomalies diffèrent en fonction de la sévérité associée. Wordcloud est un package python qui aide à visualiser les mots présents dans un texte en fonction de leur fréquence. Ci-dessous sont représentés les nuages de mots selon chacune des catégories :

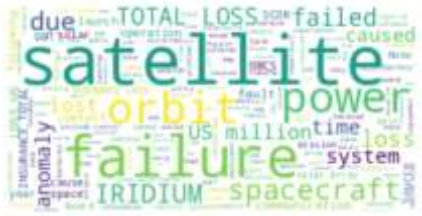
Évènement sans impact



Perte de redondance ou marge dégradée



Perte partielle



Perte totale

Figure 3.3 – Nuages de mots en fonction de la sévérité

Certains mots sont présents en forte fréquence dans chacune des catégories : c'est le cas des mots « satellite », « array », « solar » et « power ». Les évènements avec impact se distinguent des autres par des mots comme « failure » et « loss » mais il est difficile de distinguer correctement les pertes partielles, totales et pertes de redondance. Les algorithmes d'apprentissage automatique et d'apprentissage en profondeur sont formés pour aller au-delà de la simple fréquence des mots dans une description. Les algorithmes sont entraînés sur les données et prennent en compte leur disposition et les liens entre les différents mots.

Lorsqu'il s'agit d'implémenter un algorithme de machine learning, la première chose à faire est de préparer les données qui seront introduites dans l'algorithme d'apprentissage. En effet, un algorithme entraîné avec des données non prétraitées est susceptible de ne pas fonctionner correctement ou pire, de fonctionner plutôt bien à première vue tout en présentant de mauvaises capacités de généralisation. Dans la partie suivante, nous expliciterons ces étapes de prétraitement. Les composantes non pertinentes du texte seront tout d'abord retirées puis les descriptions seront vectorisées.

3.3.2 Retrait des composantes non pertinentes du texte

Quand on s'intéresse à des problèmes de classification de textes, bien souvent on n'a que les mots qui composent ces textes pour nous aider. Les actions décrites dans la suite de cette partie visent à réduire au maximum le vocabulaire utilisé dans la description des anomalies. L'objectif sera de représenter ces données sous une forme numérique afin qu'elles puissent être lisibles par les différents algorithmes d'apprentissage utilisés.

3.3.2.1 Retrait des stop-words

En observant les textes décrivant les anomalies, on se rend vite compte que certains mots se retrouvent dans plusieurs phrases mais n'ont aucune importance pour la classification d'une anomalie. Il peut s'agir notamment de prépositions, déterminants, conjonctions de coordination. On appelle ces mots des stop-words et on en dénombre environ 200 en anglais, directement accessibles via NLKT (librairie Python pour le NLP).

Pour chaque description d'anomalies, les stop-words compris dans la liste NLKT sont retirés.

Description avec stop words	Description sans stop words
Sudden increase in friction torque of Momentum Wheel leading to pitch angle deviation.	Sudden increase friction torque Momentum leading pitch angle deviation.
Automatic switch to redundant wheel.	Automatic switch redundant wheel.
Pointing transient on satellite in July caused by an increase in the friction torque of the momentum wheel.	Pointing transient satellite July caused increase friction torque momentum wheel.

Tableau 3.1 – Extrait des descriptions d'anomalies avec et sans stop words

3.3.2.2 Stemming

Une autre pratique, moins courante pour réduire un peu plus la taille du vocabulaire employé dans l'ensemble des descriptions d'anomalies est le stemming ou racinisation en français. L'opération de stemming consiste à récupérer la racine d'un mot en suivant pour la langue anglaise une liste de règles de « dessufixation ». La racine d'un mot n'est pas forcément un mot existant dans le dictionnaire. Ci-dessous sont présentés des mots provenant des descriptions d'anomalies, avant et après le processus de racinisation :

Mots	Mots après processus de stemming
Increasing	Increas
satellite	satellit
occurence	occur
estimated	estim
information	inform
provided	provid

Tableau 3.2 – Mots extraits des descriptions avant et après racinisation

3.3.3 Représentation vectorielle du texte

Une fois la réduction vocabulaire effectuée, il s'agit désormais de représenter numériquement les descriptions d'anomalies afin que les modèles de classification soient capables de les interpréter. L'objectif est de représenter chacune des descriptions de notre corpus sous forme d'un vecteur. On appelle cela la vectorisation. Il existe pour cela un grand nombre de méthodes de vectorisation. Trois d'entre elles seront utilisées dans ce projet : le bag of words, la vectorisation TF IDF et la vectorisation Word2Vec

3.3.3.1 Vectorisation bag of words

La vectorisation bag of words est la représentation la plus simple que l'on puisse concevoir. Une description est représentée par un vecteur de la taille du vocabulaire rencontré dans l'ensemble du corpus à représenter et chaque composante du vecteur mesure la fréquence d'apparition d'un mot de ce vocabulaire dans la description.

Par exemple, considérons un corpus de texte à représenter contenant les deux phrases suivantes :

- This is a database lab of this IS master course.
- This is a database course

Nous obtenons alors la base de données suivante après vectorisation :

	This	is	a	database	lab	of	IS	master	course
Phrase 1	2	1	1	1	1	1	1	1	1
Phrase 2	1	1	1	1	0	0	0	0	1

Tableau 3.3 – Données après vectorisation en bag of words

Il existe une variante un peu moins naïve de cette représentation que nous allons décrire ci-dessous : la méthode TF IDF.

3.3.3.2 Vectorisation TF IDF (term frequency - inverse document frequency)

Le premier objectif de la méthode TF IDF est de donner moins d'importance à des mots qui se retrouveraient dans l'ensemble des descriptions utilisées pour construire le vocabulaire. Il est par exemple probable que l'on retrouve le mot « issue » dans de nombreuses phrases et il serait bien de ne pas donner autant d'importance à ce mot qu'à des mots qui permettront réellement de discriminer une description d'une autre pendant la phase de classification. A contrario, la méthode TF IDF accorde plus d'importance aux mots paraissant plusieurs fois dans une même description. Ainsi, si une anomalie fait référence à une explosion et que ce mot est présent plusieurs fois dans la phrase à vectoriser, alors la méthode TF IDF lui accordera plus de poids dans la vectorisation.

Pour chaque description et pour chaque mot du vocabulaire construit, la valeur attribuée est la suivante :

$$w_{i,j} = N_{i,j} \log(\frac{N}{N_i})$$

Où $N_{i,j}$ est le nombre d'occurrences du mot i dans la description j, N est le nombre de descriptions du corpus à représenter et N_i est le nombre de descriptions contenant le mot i.

La librairie Sklearn de Python que nous allons utiliser dans le cadre de ce mémoire utilise une variante de la formule TF IDF :

$$w_{i,j} = N_{i,j}[\log\left(\frac{1+N}{1+N_i}\right) + 1]$$

Pour chaque mot i de chaque phrase j, les pondérations sont ensuite normalisées et varient dans l'intervalle [0 ;1] :

$$w_{i,j_{normalisé}} = \frac{w_{i,j}}{\sqrt{\sum_{i\prime=1}^{m_j} {w_{i\prime,j}}^2}}$$

Avec m_j le nombre de mots contenus dans la phrase j

En reprenant, l'exemple précédent, le corpus vectorisé est le suivant :

	course	database	is	lab	master	of	this
Phrase 1	0.25	0.25	0.50	0.35	0.35	0.35	0.5
Phrase 2	0.5	0.5	0.5	0	0	0	0.5

Tableau 3.4 – Données après vectorisation en TF IDF

La vectorisation TF IDF possède cependant ses limites. En effet, elle ne permet pas de capter le contexte, la similarité sémantique ou encore les relations entre les différents mots. Pour permettre de résoudre ce problème, nous allons tester une troisième et dernière méthode de vectorisation : la vectorisation word2vec.

3.3.3.3 Vectorisation word2vec

Contrairement aux vectorisations bag-of-words et TF IDF, la représentation en word2vec est issue d'un algorithme d'apprentissage automatique : le word-embedding. La représentation des mots est apprise à partir d'un corpus de textes et chaque mot est représenté par un vecteur de nombres réels. Le modèle word2vec fut implémenté par Google en 2013 sous la direction de Tomas Mikolov. Il existe un modèle pré-entraîné sur un ensemble de textes Wikipédia disponibles avec la librairie Gensim de Python. Les données de Wikipédia sont denses, volumineuses et contiennent de nombreuses informations sur le domaine des satellites. Nous l'utiliserons dans le cadre de ce mémoire.

Le modèle pré-entraîné contient en tout 3 000 000 mots référencés, chacun étant représenté par un vecteur de 300 dimensions. Le modèle word2vec est capable d'identifier le contexte et les relations entre les mots. Deux mots semblables seront représentés par des vecteurs relativement peu distants dans l'espace vectoriel où ils sont définis. Il est également possible d'effectuer des équations entre les différents mots. Si l'on effectue le calcul vectoriel $roi + femme - homme$, le vecteur obtenu le plus proche sera celui de reine :

```
GoogleModel.most_similar(positive=['king', 'woman'], negative=['man'], topn=1)
[('queen', 0.7118193507194519)]
```

Figure 3.4 – Extrait code Python : calcul vectoriel roi + femme - homme

Pour le modèle, roi est à reine, ce qu'homme est vis-à-vis de femme.

Sur le graphique ci-dessous sont représentés en 3 dimensions les 8 mots les plus proches du mot « satellite » selon le modèle pré-entraîné. Nous distinguons notamment les mots « geostationary », « NASA » et « communications » :

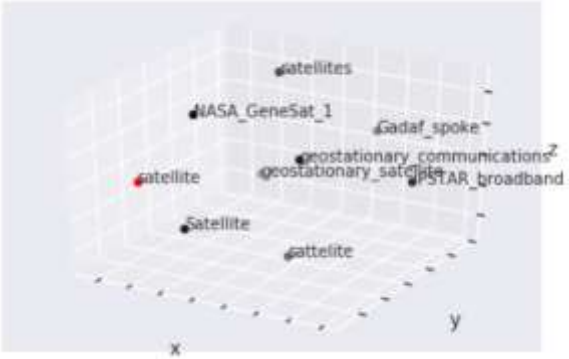


Figure 3.5 – Projection en 3D des 8 mots les plus proches de « satellite »

Le second avantage du modèle word2vec est que la taille de chaque vecteur est considérablement réduite. Là où le vecteur représentatif d'une description aura comme dimension le nombre de mots de l'ensemble du corpus avec la vectorisation bag of words et TF IDF, le modèle word2vec la représentera comme la moyenne des vecteurs des mots la composant.

Prenons un exemple simple de notre jeu de données d'entraînement : la phrase "Reaction Wheel 1 measured speed was zero despite strong torque demand." contient 10 mots, chaque mot est associé à un vecteur dans word2vec. Lorsque nous effectuons la moyenne de chacun de ces vecteurs, nous obtenons le vecteur représentatif de la phrase :

	1	**2**	**3**	**4**	**5**	**6**	**…**	**298**	**299**	**300**
reaction	0.132	-0.005	-0.018	-0.199	-0.238	0.021	…	0.098	0.083	0.178
wheel	0.137	-0.12	-0.06	0.002	-0.124	-0.007	…	0.097	-0.041	-0.268
measured	-0.159	-0.1	0.264	0.147	0.008	-0.073	…	0.008	-0.089	-0.122
speed	0.079	0.297	0.289	-0.059	0.014	0.009	…	0.002	-0.124	-0.245
was	0.026	-0.002	0.186	-0.052	0.005	-0.11	…	-0.122	0.222	-0.022
zero	0.067	-0.125	0.18	0.299	-0.189	0.275	…	0.13	-0.436	0.23
despite	0.134	0.092	-0.067	-0.149	-0.041	0.024	…	-0.026	0.287	-0.028
torque	0.389	0.082	0.048	-0.144	-0.055	-0.285	…	0.182	-0.41	-0.092
demand	-0.028	0.013	-0.032	0.154	-0.281	-0.162	…	0.043	0.215	-0.179

MOYENNE

	1	**2**	**3**	**4**	**5**	**6**	**…**	**298**	**299**	**300**
Vecteur	0.09	0.015	0.098	0.000	-0.100	-0.034	…	0.046	-0.033	-0.061

Figure 3.6 – Vectorisation Word2Vec de la phrase

Concentrons-nous désormais sur le fonctionnement et la construction d'un modèle word2vec. Avec cette méthode, le sens de chaque mot est défini par son contexte, ie par les mots qui l'entourent. Le contexte d'un mot central w_t sont les mots se trouvant entre w_{t-m} et w_t puis entre w_t et w_{t+m}, pour un rayon m fixé au départ :

Taille de la fenêtre : 9
Mot central

w_{t-4}	w_{t-3}	w_{t-2}	w_{t-1}	$\boldsymbol{w_t}$	w_{t+1}	w_{t+2}	w_{t+3}	w_{t+4}

Figure 3.7 – Contexte du mot w_t

m est un paramètre de l'algorithme qui caractérise la longueur fixe de la fenêtre et définit le contexte, ici égale à 9. La fenêtre représente la dimension des vecteurs des mots du modèle entraîné. Dans le modèle word2vec pré-entraîné utilisé, chaque vecteur est de dimension 300, qui est également la longueur de la fenêtre utilisée.

Il existe deux types d'algorithme de word-embedding :

- Le modèle continuous bag of words qui permet de prédire un mot à partir de son contexte
- Le modèle skip gram qui lui permet de prédire le contexte à partir d'un mot

Fonctionnement de l'algorithme :

Nous allons nous concentrer ici sur le fonctionnement du modèle skip gram qui est implémenté dans le modèle pré-entraîné utilisé.

Soit un corpus de textes possédant un ensemble de vocabulaires de mots de taille V. Le fonctionnement du modèle Word2Vec repose sur deux matrices de projection W_S et W_C de dimensions (n_V, N) où N est la dimension de la projection que l'on choisit (300 dans notre cas). Pour chaque mot w_t du vocabulaire, la $t^{ème}$ ligne de la matrice W_S est la représentation du mot dans l'espace des sens et la $t^{ème}$ ligne de la matrice W_C est sa représentation dans l'espace des contextes. On notera par la suite les deux vecteurs associés w_t (S) et w_t (C).

Considérons deux mots du vocabulaire notés w_u et w_v. La probabilité que w_u appartienne au contexte de w_v se définit comme suit (fonction softmax ou exponentielle normalisée) :

$$P(w_u | w_v) = \frac{\exp[\, w_u\,(\mathrm{C})^T * w_v(\mathrm{S})\,]}{\sum_{i=1}^{V} \exp[\, w_i\,(\mathrm{C})^T * w_v(\mathrm{S})\,]}$$

L'objectif général du modèle word2vec est de maximiser la vraisemblance définit par :

$$L(\theta) = \prod_{i=1}^{V} \prod_{w_u \in \mathrm{C}(w_i)} P(w_u | w_v, \theta = (W_S, W_C))$$

Avec $C(w_i)$ étant le contexte du mot w_i. Les paramètres du modèle sont les deux matrices :

$$\theta = (W_S, W_C)$$

Les matrices W_S et W_C sont tout d'abord initialisées aléatoirement. Les poids sont ensuite ajustés au fur et à mesure de la phase d'apprentissage sur le principe du réseau de neurones et de la descente de gradient. Le fonctionnement général des réseaux de neurones sera expliqué dans la partie 4.4.6 de ce mémoire.

3.4 Présentation des modèles utilisés

Comme évoqué précédemment, les quatre modèles utilisés pour la classification en sévérité sont les suivants :

- SVM
- Random Forest
- Gradient Boosting
- Réseau de neurones

Dans cette partie, nous allons expliquer le fonctionnement de ces algorithmes et leur application dans une classification multiclasse.

3.4.1 SVM

Introduit par Vapnik en 1998, le classifieur SVM est un algorithme d'apprentissage automatique supervisé qui consiste à classer des données grâce à des séparateurs linéaires. Ces séparateurs peuvent être des vecteurs ou bien des hyperplans dans un espace de plus grande dimension.

Supposons que nous faisons face à un problème de classification binaire. Soit un ensemble d'apprentissage $D = \{(x^{(i)}, y^{(i)}) : x^{(i)} \in \mathbb{R}^d, y^{(i)} \in \{-1, +1\}\}_{i=1,\dots,n}$. L'objectif du modèle SVM est de chercher une fonction $f: \mathbb{R}^d \to \mathbb{R}$ qui permet de prédire si une observation $x \in \mathbb{R}^d$ est de classe −1 ou +1.

Séparateur à vaste marge :

Supposons que les classes -1 et +1 sont séparables par un hyperplan. Considérons un vecteur x à d composantes numériques $x = (x_1, \dots, x_d)$. Un hyperplan H d'équation $f(x)$ a la forme :

$$f(x) = \sum_{i=1}^{d} w_i x_i + b = < w, x > + b$$

avec w le vecteur orthogonal à l'hyperplan et b le déplacement par rapport à l'origine. $< ., . >$ est le produit scalaire définit comme suit : $< x, y > = \sum_{i=1}^{d} x_i * y_i$.

Afin de déterminer l'hyperplan optimal pour séparer les deux classes, nous utilisons la marge :

$$Marge(H) = \min_{x^{(i)}} d(x^{(i)}, H)$$

L'hyperplan optimal du modèle SVM est celui qui maximise la marge, d'où le nom de « séparateur à vaste marge ». L'objectif est de maximiser la séparation des points de données à leurs deux classes potentielles.

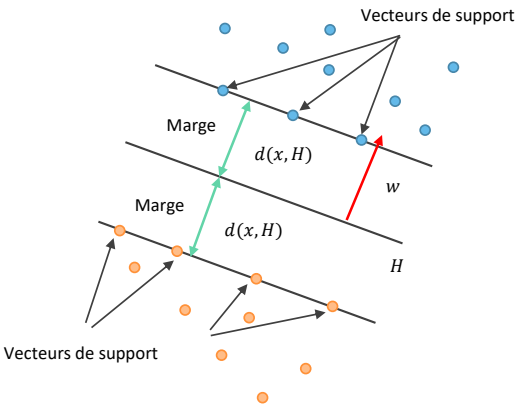


Figure 3.8 – Séparateur à vaste marge pour une classification binaire

Sur le schéma ci-dessus, les exemples d'apprentissage appartenant à la classe 1 les plus proches de l'hyperplan se trouvent à la même distance de l'hyperplan que les exemples d'apprentissage les plus proches de classe -1. Ces éléments sont appelés « vecteurs de support ».

Après avoir déterminé l'hyperplan $f(x)$, la classification d'une nouvelle observation se fait de la façon suivante :

- Si $f(x) = 0$: l'élément se trouve sur la frontière de séparation.
- Si $f(x) > 0$: on prédit l'élément comme étant de classe 1.
- Si $f(x) < 0$: on prédit l'élément comme étant de classe -1.

SVM linéaire (cas séparable) :

Supposons qu'il existe un hyperplan qui sépare les exemples d'apprentissage sans erreur. Il a pour équation :

$$f(x) = < w, x > + b = w^T x + b$$

Soit $x^{(s)}$ un vecteur de support et l'ensemble H d'équation $H = \{x | w^T x + b = 0\}$ (nous considérons que le seuil de décision est 0), alors la marge se calcule comme suit :

$$Marge = 2d\left(x^{(s)}, H\right) = 2\,\frac{|\, w^T x^{(s)} + b \,|}{||w||}$$

Cette quantité vaut deux fois la marge (par rapport à la définition précédente).

Posons ensuite la condition de normalisation : $|\, w^T x^{(s)} + b \,| = 1$. Elle amène à la formule suivante :

$$Marge = \frac{2}{||w||}$$

Si $x^{(i)}$ est un vecteur de support appartenant à la classe +1, alors $w.x^{(i)} + b = 1$ et si $x^{(i)}$ est de classe -1, alors $w.x^{(i)} + b = -1$. Nous avons donc les équations suivantes :

$$\begin{cases} w.x^{(i)} + b \geq 1 \; si \; y^{(i)} = 1 \\ w.x^{(i)} + b \leq -1 \; si \; y^{(i)} = -1 \end{cases} \Rightarrow y^{(i)} \, (w.y^{(i)} + b) \geq 1$$

Le problème d'optimisation est ainsi :

$$\begin{cases} \min\limits_{w,b} \frac{1}{2} ||w||^2 \\ tel \; que \; y^{(i)} \, (w.x^{(i)} + b) \geq 1, i = 1, \dots, n \end{cases}$$

La résolution de ce système par la méthode des multiplicateurs de Lagrange fournit l'équation de l'hyperplan optimal :

$$f(x) = \sum_{i=1}^{n} \lambda_i{}^* y^{(i)} \, x^{(i)^T} \, x + b^* = 0$$

Avec λ_i les multiplicateurs de Lagrange.

Classes non linéairement séparables :

La séparation des exemples d'apprentissage se fait rarement sans erreur de classification. Le SVM linéaire ajoute alors une variable d'ajustement ξ dans le problème d'optimisation :

$$y^{(i)}\left(w.x^{(i)} + b\right) \geq 1 - \xi^{(i)} \text{ avec } i \in \{1, \dots, n\}$$

Le vecteur x n'est pas correctement classifié lorsque sa variable d'ajustement ξ est supérieure à 1. Nous sommes ramenés alors au problème de minimisation suivant :

$$\begin{cases} \min\limits_{w,b} \frac{1}{2} ||w||^2 + C \sum\limits_{i=1}^{n} \xi^{(i)} \\ tel \; que \; y^{(i)}\left(w.x^{(i)} + b\right) \geq 1 - \xi^{(i)}, i = 1, \dots, n \\ \xi_i \geq 0, i = 1, \dots, n \end{cases}$$

C est le paramètre de régularisation. Il permet de trouver un compromis entre les erreurs de classification et la largeur de la marge.

Comme précédemment, le système est résolu par la méthode du Lagrangien. L'équation de l'hyperplan permettant de classer une nouvelle donnée x est toujours :

$$f(x) = \sum_{i=1}^{n} \lambda_i{}^* y^{(i)} < x^{(i)}, x > + b^* = 0$$

Les séparateurs non linéaires :

Lorsque la séparation des données ne peut pas s'effectuer de manière linéaire, une fonction Φ est utilisée. Elle permet de projeter les données de l'espace $\mathbb{R}^d$ dans un espace de Hilbert $\mathcal{H}$ de plus grande dimension. Dans ce nouvel espace, une séparation linéaire des données pourra être effectuée. Cette méthode s'appelle la méthode du noyau et utilise les produits scalaires donnés par une fonction noyau :

$$K\left(x^{(i)}, x^{(j)}\right) = \left(\Phi\left(x^{(i)}\right), \Phi\left(x^{(j)}\right)\right)$$

Le schéma suivant illustrant l'astuce des noyaux est issu de la thèse [19].

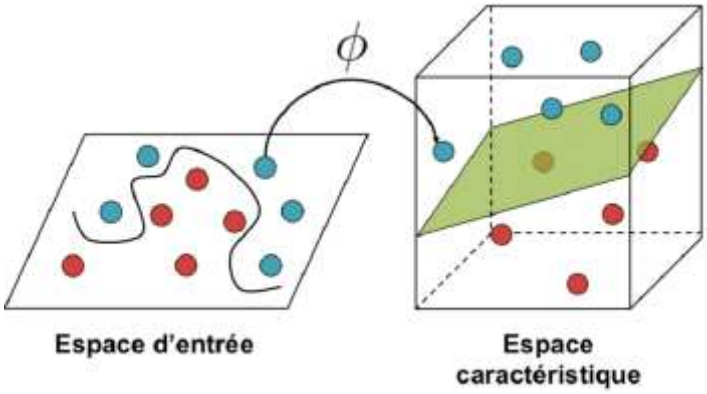


Figure 3.9 - Schéma illustrant l'astuce des noyaux

Toute fonction K ne peut pas être noyau. La fonction K doit être continue, symétrique et semi définie positive. Voici la liste non exhaustive des noyaux couramment utilisés :

- Le noyau polynomial de degré p ⇒ $K(x^{(i)}, x^{(j)}) = (1 + x^{(i)}.x^{(j)})^p$
- Le noyau gaussien ou noyau radial ⇒ $K(x^{(i)}, x^{(j)}) = exp(-\frac{\|x^{(i)}-x^{(j)}\|^2}{2\sigma^2})$ avec $\sigma > 0$.
- Le noyau tangente hyperbolique ⇒ $K\left(x^{(i)}, x^{(j)}\right) = tanh(\alpha x^{(i)}.x^{(j)} + \beta)$ avec $\alpha, \beta \in \mathbb{R}$.

SVM pour classification multiclasse :

Il existe plusieurs méthodes de classification multiclasse. Celle retenue pour la modélisation est la méthode OneVsRest. Pour chaque classe k, un modèle SVM binaire est appris via les données $(x^{(i)}, 1_{\{y^{(i)}=k\}})$ avec $x^{(i)}$ les variables explicatives et $y^{(i)}$ la variable à expliquer. Lors de la prédiction, l'observation est affectée à la classe qui maximise le score de classification.

Le schéma ci-dessous présente la méthode OneVsRest : un hyperplan est construit pour séparer une classe de toutes les autres à la fois. Pour la classe verte, la ligne associée maximise la séparation entre les points verts et les autres points.

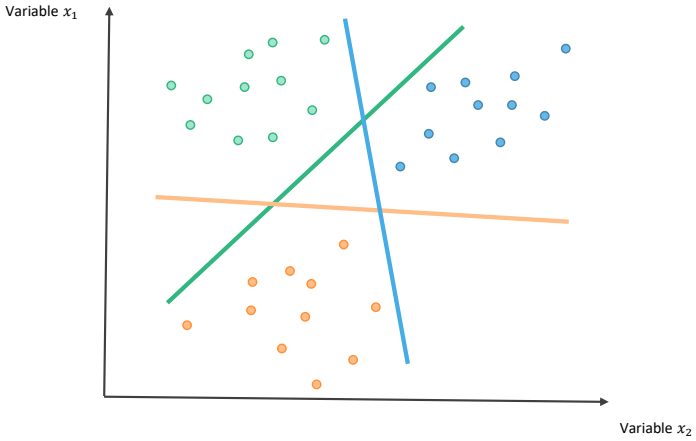


Figure 3.10 - SVM dans le cas d'une classification multiclasse

3.4.2 Arbre de décision

Les arbres de décision sont des algorithmes itératifs qui permettent de prédire une variable continue ou discrète. A chaque étape, les individus sont séparés en n classes (le plus souvent n = 2) selon une des variables explicatives. Lors de cette séparation, la variable est choisie de telle sorte que l'on ait la meilleure division possible. Les sous-groupes obtenus à partir de cette division correspondent aux nœuds de l'arbre. Le nœud initial de l'arbre correspond à l'ensemble de l'échantillon et une séparation est effectuée sur chacun des sous-ensembles. Les feuilles de l'arbre correspondent aux derniers nœuds et contiennent la classe à prédire dans le cas d'une classification ou la valeur cible dans le cas d'une régression.

Les éléments suivants sont nécessaires à la construction d'un arbre de décision :

- Le choix d'un critère permettant la sélection de la meilleure division parmi toutes celles disponibles pour l'ensemble des variables explicatives.
- La définition d'une règle permettant de déterminer l'arrêt des itérations et le choix du nœud terminal.
- L'affectation de chaque feuille à l'une des classes pour la prédiction d'une variable discrète ou à une valeur pour la prédiction d'une variable continue.

Dans le cas d'une classification, l'objectif est de choisir à chaque nœud une variable accordant le maximum d'homogénéité dans la division. On atteint un maximum d'homogénéité lorsque tous les individus sont répartis sur le même mode. On dit aussi que le nœud est pur. Le choix du critère de sélection de la meilleure division se base sur l'indice de Gini. Il permet de mesurer le pouvoir discriminant d'une variable et est calculé comme suit :

$$i(t) = \sum_{k=1}^{K} p_{t,k}\left(1 - p_{t,k}\right) = 1 - \sum_{k=1}^{K} p_{t,k}^2$$

Avec t le numéro de la variable considérée et $p_{t,k}$ la proportion d'observations appartenant à la classe k de la variable.

L'indice de Gini est compris entre 0 (lorsque la répartition entre les classes est homogène) et 1 (lorsqu'on atteint un maximum d'hétérogénéité, les classes de la variable sont équiprobables). La variable sélectionnée pour le nœud considéré sera celle possédant l'indice de Gini le plus faible.

Voici ci-dessous le schéma complet d'un arbre binaire à un niveau :

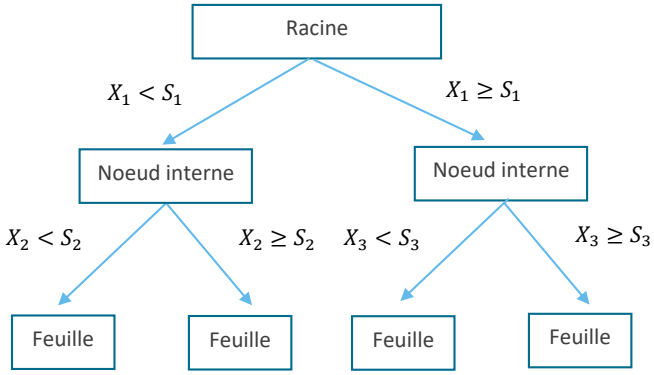


Figure 3.11 - Arbre binaire à un niveau

A chaque nœud de l'arbre, la variable X_i est testée selon un seuil s_i. Une variable peut être sélectionnée plusieurs fois dans le même arbre. Dans le cas d'une classification, chacune des feuilles correspond à une classe précise.

3.4.3 Random forest

Le modèle random forest est une méthode d'apprentissage ensembliste permettant de résoudre un problème de régression ou de classification. Le premier algorithme fut proposé en 1995 par Tin Kam Ho. Une extension fut ensuite développée par Leo Breiman et Adele Cutler en 2001. Un modèle de random forest fait des prédictions basées sur un certain nombre de modèles de faible qualité individuelle. En combinant ces modèles individuels, le modèle d'ensemble a tendance à être plus flexible (moins de biais) et moins sensible aux données (moins de variance). Les modèles individuels sont alors appelés apprenants faibles (ou weak learners) et le modèle final est l'apprenant fort (ou strong learner). Cette technique se nomme le bagging.

Le random forest est composé d'un ensemble d'arbres de décision indépendants et chaque arbre est créé à partir d'un double tirage aléatoire :

- Un tirage aléatoire avec remplacement sur les observations que l'on appelle le tree bagging
- Un tirage aléatoire avec remplacement sur les variables explicatives : le feature sampling

Dans le cas d'une classification, la prédiction faite par le random forest pour des données inconnues est le vote à la majorité des arbres. Dans le cas d'une régression, il s'agit de la moyenne des résultats. Le fonctionnement de l'algorithme est résumé sur le schéma ci-dessous :

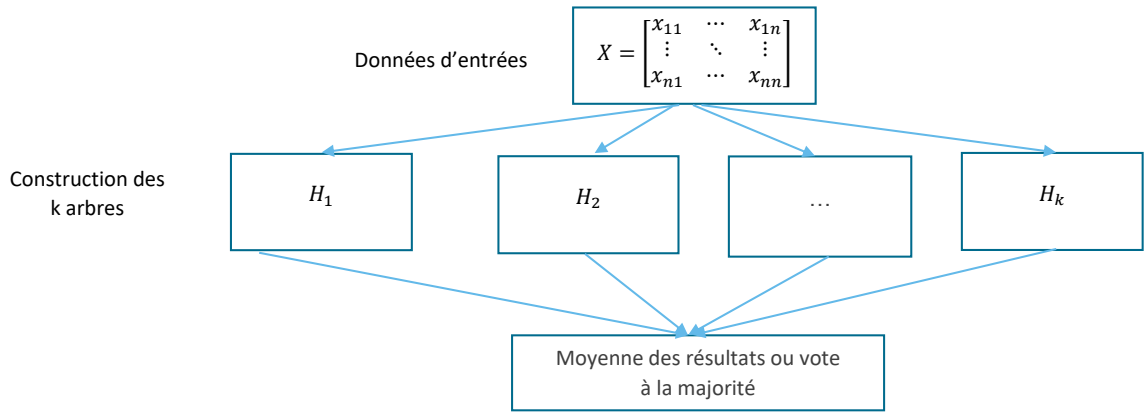


Figure 3.12 - Processus d'une forêt aléatoire

3.4.4 Introduction au boosting avec l'algorithme adaboost

Tout comme la technique de bagging utilisée dans le random forest, le boosting est une méthode d'apprentissage ensembliste améliorant la performance d'un modèle. Cependant, au lieu de considérer les weak learners indépendamment les uns des autres, le modèle les apprend séquentiellement de manière très adaptive (le modèle de base dépend des précédents) et les combine selon une stratégie déterministe.

Adaboost fut l'un des premiers algorithmes de boosting mis au point par Freund & Schapire en 1996. Son principe est le suivant :

- Un premier arbre de décision appelé stump est d'abord entraîné. Cet arbre contient seulement deux feuilles et sa simplicité l'empêche d'être performant.
- Un second arbre est ensuite créé en tenant compte du précédent : les observations mal classées ont un poids plus important dans l'apprentissage que ceux qui ont été bien classées. En réitérant ce processus, une suite d'arbres est ainsi créée, chaque arbre compensant la faiblesse du précédent.
- Dans le cas d'une classification, le modèle final prédit la classe ayant obtenu le plus de votes par les weak learners. Contrairement au random forest, les arbres n'ont pas tous le même poids dans la prédiction. La pondération sera plus élevée pour les arbres ayant le mieux performé.

Supposons que l'on soit dans le cas d'une classification à deux classes (-1 ou 1). Pour chaque observation x, la prédiction finale est la suivante :

$$H_{final}(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$$

Avec :

- T le nombre de classifieurs
- $h_t(x)$ le résultat du classifieur n°t
- α_t la pondération associée au classifieur n°t

3.4.5 Gradient Boosting

L'algorithme de Gradient Boosting fut développé par le statisticien américain Jerome Friedman dans les années 1990. Cette méthode d'apprentissage automatique combine deux méthodes : la méthode du boosting et la méthode de descente de gradient.

Soient les données d'une variable $y = f(x)$ que l'on souhaite prédire et p variables explicatives sous la forme d'un vecteur x. Un échantillon $\{(x_1, y_1), \dots, (x_n, y_n)\}$ de n observations est donné. L'objectif de l'algorithme de gradient boosting est de minimiser une fonction de perte L. Dans le cas d'une classification, la fonction de perte se calcule comme suit :

$$L\big(y, f(x)\big) = \begin{cases} 0 \; si \; y = f(x) \\ 1 \; sinon \end{cases}$$

On appelle fonction d'erreur l'espérance mathématique de la fonction de perte :

$$R(f) = E[L(y, f)]$$

L'objectif du modèle de gradient boosting est de trouver une approximation $\hat{f}$ de la fonction f^* qui minimise cette fonction d'erreur :

$$f^*(x) = \; argmin_{\mathrm{f}} \, E(L\big(y, f(x)\big)]$$

Afin d'approximer cette fonction f^*, le gradient boosting opte pour un algorithme qui vise à appliquer la descente de gradient et utiliser les pseudo résidus des prédictions de la base de données d'entraînement à chaque itération. Les différentes étapes de l'algorithme sont résumées à la page suivante :

Etape 1 : Initialisation du modèle avec une constante :

$$f_0(x) = argmin_\Upsilon \sum_{i=1}^{n} L(y_i, \Upsilon)$$

Etape 2 : Pour m allant de 1 à M :

- Calcul des pseudo-résidus $r_{i,m} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=f_{m-1}(x)}$

- Construction d'un weak learner h_m (arbre de décision) en l'entraînant sur les données $(x_i, r_{i,m})_{i=1}^{n}$

- Calcul du poids associé à ce weak learner grâce à l'équation :

$$\Upsilon_m = argmin_\Upsilon \sum_{i=1}^{n} L(y_i, f_{m-1}(x_i) + \Upsilon h_m(x_i))$$

- Mise à jour du modèle : $f_m(X) = f_{m-1}(X) + \alpha * \Upsilon_m * h_m(X)$

  Le coefficient α est appelé le taux d'apprentissage de l'algorithme et correspond à la pondération de chaque arbre de décision dans l'équation de la prévision.

Etape 3 : Le modèle final est le suivant : $f_M(X)$

3.4.6 Les réseaux de neurones

Le quatrième et dernier modèle construit dans le cadre de la classification s'appuie sur un réseau de neurones. Le réseau de neurones artificiels est l'un des algorithmes les plus utilisés de l'apprentissage automatique et peut être utilisé pour des tâches de régression ou de classification. L'algorithme permet notamment la résolution de problèmes complexes tels que la vision par ordinateur ou le traitement du langage naturel.

Le fonctionnement des réseaux de neurones est basé sur l'apprentissage profond (ou deep learning) dont le concept fut inventé en 1943 par les chercheurs américains Warren McCullough et Walter Pitts. Cet algorithme s'inspire du fonctionnement des réseaux de neurones naturels et est constitué de plusieurs couches de nœuds. Il contient au moins trois couches :

- La couche d'entrée qui contient autant d'unités que de variables explicatives
- Les couches suivantes sont appelées couches cachées. Il peut y avoir une ou plusieurs couches cachées, d'où le terme de deep learning.

- La couche de sortie qui contient un seul neurone lorsque la variable à expliquer est continue ou lors d'une classification binaire. Dans le cas d'une classification multiclasse, il y a autant de neurones de sortie que de classes.

Sur le schéma ci-dessous est représentée la structure d'un réseau de neurones constitué d'une unique couche cachée. Dans cet exemple, la couche d'entrée contient trois neurones et la couche de sortie deux neurones :

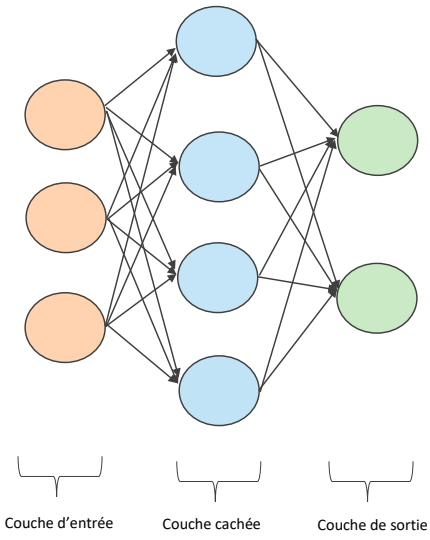


Figure 3.13 - Réseau de neurones contenant une seule couche cachée

Les neurones ont des fonctions différentes selon la couche à laquelle ils appartiennent. Les neurones de la première couche transfèrent les données d'entrée vers la première couchée cachée. Les neurones situés dans les couches cachées et dans la couche de sortie ont des fonctions similaires :

- Tout d'abord, chaque neurone résume les données de la couche précédente en effectuant le calcul suivant : $\sum_{i=1}^{N} n_i * p_i$ avec N le nombre de neurones de la couche précédente, n_i est la valeur retour du nœud i de la couche précédente et p_i son poids accordé.
- Ensuite, le neurone applique une fonction d'activation f et un biais B aux données résumées. La valeur de sortie de la couche est donc : $f(\sum_{i=1}^{m} n_i p_i + B)$. La fonction d'activation est utilisée pour introduire une non-linéarité dans la sortie d'un neurone et permet de normaliser toutes les sorties des nœuds du réseau sur une plage finie de valeur.

Le principe du fonctionnement d'un neurone d'une couche cachée ou de sortie est résumé par le schéma suivant :

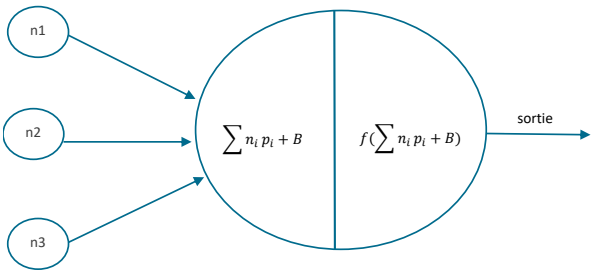


Figure 3.14 - Principe de fonctionnement d'un neurone

Considérons un réseau à L couches, pour la suite des analyses les notations suivantes seront utilisées:

- $w^l = \begin{pmatrix} w_{1,1}^l & \cdots & w_{1,J}^l \\ \vdots & \ddots & \vdots \\ w_{I,1}^l & \cdots & w_{I,J}^l \end{pmatrix}$ la matrice des poids reliant la couche $l-1$ (comportant J neurones) à la couche l (comportant I neurones)
- $W = (w^1, \dots, w^{L-1})$ l'ensemble des poids du réseau
- $b^l = (b_1^l, \dots, b_I^l)$ les biais de la couche l
- $B = (b^2, \dots, b^L)$ l'ensemble des biais des couches cachées et de la couche de sortie
- $a^l = (a_1^l, \dots, a_I^l)$ l'ensemble des sorties de la couche l
- $z^l = (z_1^l, \dots, z_I^l)$ l'ensemble des entrées de la couche l

Dans le cas d'une classification multiclasse, la fonction softmax est appliquée comme fonction d'activation sur la dernière couche du réseau. Elle permet de transformer un vecteur réel en vecteur de probabilités et est définie comme suit :

$$softmax(z_i^L) = \frac{e^{z_i^L}}{\sum_{j=1}^{J} e^{z_j^L}}$$

On interprète la valeur $softmax(z_i^L)$ comme étant la probabilité d'appartenance d'une observation à la $i^{ème}$ classe.

Pour entraîner le modèle, il est nécessaire de définir une fonction qui va quantifier l'erreur de prédiction sur l'ensemble d'apprentissage. En général, pour les problèmes de classification, la fonction « cross - entropy » est utilisée. Elle est définie comme suit :

$$E_j = -\sum_{c=1}^{C} y_j^c * \log(p_j^c)$$

Avec :

- j : le numéro de l'observation
- C : le nombre total de classes
- y_j^c valant 0 si l'observation j n'appartient pas à la classe c et 1 sinon
- p_j^c : la probabilité d'appartenance de l'observation j à la classe c

E_j varie dans l'intervalle $[0,1]$, où 0 correspondrait au modèle parfait. Le problème d'optimisation consiste à trouver les poids qui minimisent cette fonction d'erreur.

Tout comme pour le modèle de gradient boosting, la méthode de descente de gradient est utilisée. Les poids sont tout d'abord initialisés de manière aléatoire. L'objectif est de suivre l'inverse du gradient de la fonction d'erreur à chaque nouvelle itération en modifiant la valeur des poids. L'algorithme suit les étapes ci-dessous :

- Initialiser W_0 et B_0 l'ensemble des poids et des biais du modèle de façon aléatoire
- A chaque itération :
    - Tirer aléatoirement un échantillon d'apprentissage de taille m (appelé batch)

- Calculer les erreurs de prédiction des observations appartenant à ce batch à l'aide de la fonction de perte puis ajuster W_t et B_t :

$$W_{t+1} = W_t - \alpha * \frac{1}{m} \sum_{j=1}^{m} \frac{\partial E_j}{\partial W_t}$$

$$B_{t+1} = B_t - \alpha * \frac{1}{m} \sum_{j=1}^{m} \frac{\partial E_j}{\partial W_t}$$

Avec :

- W_t : les poids du modèle à l'itération t
- B_t : les biais du modèle à l'itération t
- α : le pas du gradient (ou vitesse d'apprentissage du modèle)
- E_j : la fonction d'erreur de la j-ième observation appartenant au batch

Les calculs des différentes dérivées partielles sont réalisés de manières successives en utilisant la méthode de la rétropropagation. Les poids et biais sont mis à jour successivement de la dernière couche vers la première grâce aux quatre équations ci-dessous qui peuvent être démontrées en utilisant le théorème de dérivation des fonctions composées :

- $\delta_i^L = \frac{\partial a_i^L}{\partial z_i^L} * \frac{\partial E}{\partial a_i^L}$ pour la couche de sortie L
- $\delta_i^l = \frac{\partial a_i^l}{\partial z_i^l} * \sum_j w_{ji}^{l+1} * \delta_i^{l+1}$ pour les couches cachées l du réseau
- $\frac{\partial E}{\partial w_{ij}^l} = a_i^{l-1} * \delta_i^l$ pour les poids
- $\frac{\partial E}{\partial b_i^l} = \delta_i^l$ pour les biais

<u>Gradient vanishing et fonction d'activation des couches cachées :</u>

La performance d'un réseau diffère en fonction de sa fonction d'activation. Lors de la mise en place des premiers réseaux de neurones, deux fonctions d'activation étaient couramment utilisées :

- La fonction sigmoïde

$$f(x) = \frac{e^x}{1 + e^x}$$

- La fonction tangente hyperbolique

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Ces fonctions varient dans des intervalles à faibles valeurs ($]0,1[$ pour la fonction sigmoïde et $]-1,1[$ pour la fonction tangente hyperbolique). Cela peut engendrer un phénomène appelé « gradient vanishing ». Lors de chaque itération d'apprentissage, chacun des poids du réseau de neurones est mise à jour proportionnellement à la dérivée partielle de la fonction d'erreur par rapport au poids actuel. Pour une fonction d'activation avec un intervalle à faibles valeurs comme

ensemble d'arrivée, le théorème de dérivation des fonctions composées lors de la rétropropagation montre que le gradient des couches proches de l'entrée peut devenir extrêmement faible, empêchant efficacement les poids de changer de valeur, le réseau ne peut donc plus apprendre.

Pour résoudre ce problème, une nouvelle fonction d'activation a été introduite en 2012 : la fonction ReLU (Rectified Linear Unit). Elle est définie comme suit :

$$f(x) = \max(0, x)$$

La fonction ReLU n'a pas de borne supérieure. Ainsi, son gradient vaut zéro lorsque x est négatif. Cela laisse alors le neurone dans un « état de mort ». Il est cependant peu probable que l'ensemble des neurones du réseau se retrouvent dans cet état. Cette fonction d'activation est aujourd'hui couramment utilisée dans les modèles des réseaux et nous allons l'utiliser dans les couches cachées de notre modèle.

3.5 Résultats et comparaisons des modèles

Après avoir présenté les méthodes de prétraitement des données textuelles et les différents algorithmes de classification, nous allons construire les modèles sur la base des données d'apprentissage. Toutes les étapes nécessaires à la construction et à la sélection du modèle le plus performant pour la classification de la sévérité des anomalies seront décrites ci-dessous.

3.5.1 Sur et sous apprentissage

Avant d'entamer la construction d'un modèle de machine learning, il est important de connaitre les différents éléments et phénomènes pouvant perturber la qualité des prédictions. Parmi eux se trouvent le sur et sous-apprentissage.

Sous-apprentissage (ou underfitting) :

Le sous-apprentissage désigne le fait que le modèle ne s'ajuste pas suffisamment aux données d'entraînement. Autrement dit, le modèle n'arrive pas à capturer les relations entre les différentes variables. Ainsi, l'erreur lors de l'apprentissage est élevée et le modèle ne pourra pas généraliser le phénomène en question sur de nouvelles observations.

Sur apprentissage (ou overfitting) :

Lors du phénomène d'overfitting, la fonction prédictive créée s'adapte trop aux données d'apprentissage. Ainsi, le modèle apprend tous les détails et le bruit présent dans la base d'entraînement. Ce phénomène va provoquer une erreur d'estimation élevée face à de nouvelles observations dont le bruit est aléatoirement différent.

Le modèle sélectionné lors de l'apprentissage de l'algorithme ne doit pas souffrir de sur ou sous apprentissage.

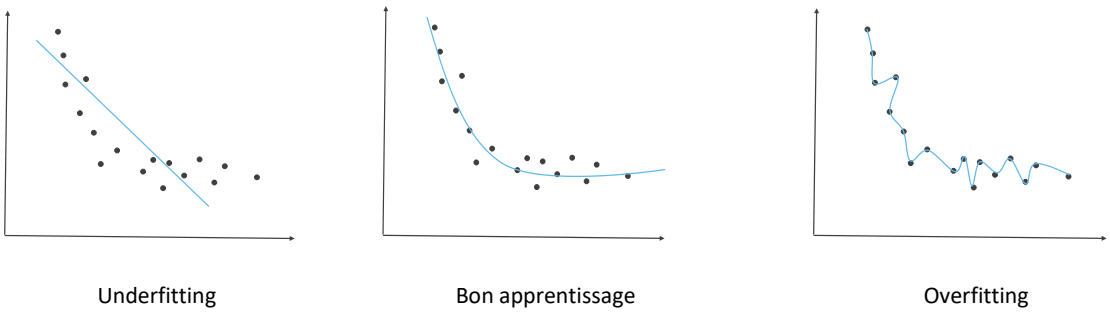


Figure 3.15 - Différences d'apprentissage pour un modèle continue

Il existe plusieurs méthodes pour empêcher ces deux phénomènes dont notamment la validation croisée que nous allons effectuer par la suite. Il existe également différents moyens de mesurer la performance d'un modèle de machine learning et de vérifier s'il ne souffre pas d'underfitting ou d'overfitting.

3.5.2 Mesures d'évaluation de la qualité des modèles

La qualité d'un modèle est mesurée à l'aide de métriques d'évaluation. Elles peuvent être appliquées aussi bien pour la classification binaire que pour la classification multiclasse. Toutes ces mesures sont calculées à partir de la matrice de confusion.

3.5.2.1 Matrice de confusion

La matrice de confusion permet de mesurer la qualité d'un modèle de classification. Elle rassemble les prévisions du modèle et les compare aux valeurs réelles. Elle se présente sous la forme suivante :

Prédictions/Observations	y_1	y_2	…	y_j	…	y_{K-1}	y_K
1	$m_{1,1}$	$m_{1,2}$	…	$m_{1,j}$	…	$m_{1,K-1}$	$m_{1,K}$
2	$m_{1,1}$	$m_{2,2}$	…	$m_{2,j}$	…	$m_{2,K-1}$	$m_{2,K}$
…	…	…	…	…	…	…	…
j	$m_{j,1}$	$m_{j,2}$	…	$m_{j,j}$	…	$m_{j,K-1}$	$m_{j,K}$
…	…	…	…	…	…	…	…
$K-1$	$m_{K-1,1}$	$m_{K-1,2}$	…	$m_{K-1,j}$	…	$m_{K-1,K-1}$	$m_{K-1,K}$
K	$m_{K,1}$	$m_{K,2}$	…	$m_{K,j}$	…	$m_{K,K-1}$	$m_{K,K}$

Tableau 3.5 – Matrice de confusion multiclasse

N est le nombre de classes et $m_{i,j}$ est le nombre d'observations ayant pour classe prédite y_i et pour classe réelle y_j.

Plus les valeurs sur la diagonale principale de la matrice sont élevées, plus le nombre d'observations correctement classées est important et meilleur est le modèle.

3.5.2.2 Erreur de prédiction (accuracy)

L'accuracy d'un algorithme de classification est un moyen de mesurer la fréquence à laquelle l'algorithme classe correctement un ensemble d'observations. Elle se calcule par le rapport entre le nombre d'observations correctement classifiées sur le nombre total d'observations :

$$accuracy = \frac{1}{n}\sum_{i=0}^{n-1} 1_{\widehat{y_i}=y_i}$$

Avec :

- n : le nombre d'observations de l'échantillon
- $\widehat{y_i}$: la classe prédite de la $i^{ème}$ observation
- y_i : la classe réelle de la $i^{ème}$ observation

Par sa facilité de calcul et d'interprétation, l'accuracy est une métrique couramment utilisée pour évaluer la performance d'un modèle de classification. En revanche, dans le cas d'un jeu de données d'entraînement déséquilibré, l'accuracy ne se révèle pas être la meilleure mesure à utiliser. Par exemple, en présence d'un jeu de données avec 10% des individus positifs et 90% négatifs, il suffit de prédire tous les individus comme négatifs pour atteindre une accuracy de 90%, ce qui est généralement considéré comme une très bonne performance. Cette métrique est donc à utiliser avec précaution.

3.5.2.3 Rappel ou sensibilité

Le rappel est la proportion d'observations correctement prédites positivement sur l'ensemble des observations réellement positives.

$$rappel = \frac{TP}{TP + FN}$$

Si l'on reprend l'exemple précédent avec un jeu de données avec 10% des individus positifs, 90% négatifs et une prédiction de tous les individus comme étant négatifs, nous obtenons le résultat suivant :

$$rappel = \frac{0}{0 + 0.1} = 0$$

La mesure de performance de notre algorithme par cette métrique donne un résultat nul. Le rappel est en fait utilisé pour mesurer le coût des faux négatifs. Lorsque l'on cherche à limiter le nombre de faux négatifs, comme dans le cas de la détection d'une maladie, c'est cet indicateur que l'on doit maximiser.

Soit N le nombre de classes du modèle et $rappel_i$ la mesure du rappel associé à la classe n°i. La généralisation au cas multiclasse de cette mesure est définie comme suit :

$$rappel_{multi\ classe} = \frac{\sum_{i=1}^{N} rappel_i}{N}$$

3.5.2.4 Précision

La précision est la proportion d'observations correctement prédites positivement sur toutes les observations prédites positivement. Elle varie donc entre 0 et 1 et se calcule comme suit :

$$précision = \frac{TP}{TP + FP}$$

A l'inverse du rappel, la précision est utilisée pour mesurer le coût des faux positifs. Lorsque l'on cherche à limiter le nombre de faux positifs, comme dans le cas de la détection de spams, c'est cet indicateur que l'on doit maximiser.

Pour généraliser cette métrique dans notre cas multiclasse, on calcule la moyenne du score précision de chaque classe individuelle :

$$précision_{multi\ classe} = \frac{\sum_{i=1}^{N} précision_i}{N}$$

3.5.2.5 Spécificité

La spécificité est la proportion d'observations correctement prédites négativement parmi toutes celles qui devraient être prédites négativement :

$$spécificité = \frac{TN}{FP + \mathrm{TN}}$$

La généralisation au cas multi classe se fait comme suit :

$$spécificité_{multi\ classe} = \frac{\sum_{i=1}^{N} spécificité_i}{N}$$

3.5.2.6 F1 score

Il est difficile d'évaluer la performance d'un modèle de classification en considérant les métriques de rappel et de précision séparément. En effet :

- Si toutes les prédictions sont positives, le rappel sera élevé
- Au contraire, si toutes les prédictions sont négatives, la précision sera élevée

Le F1 score permet de combiner la précision et le rappel. Il peut être interprété comme une moyenne pondérée de ces deux mesures. Il varie entre 0 et 1, atteint sa meilleure valeur à 1 et son pire score à 0.

$$F1\ score = 2 * \frac{précision * rappel}{précision + rappel}$$

Le $F1\ score_{multi\ classe}$ est calculé comme ceci :

$$F1\ score_{multi\ classe} = \frac{\sum_{i=1}^{N} F1\ score_i}{N}$$

3.5.2.7 Courbe ROC

Dans le cadre d'une classification binaire, la courbe ROC (Receiver Operating Characteristic) est le graphique du taux de vrais positifs (taux de sensibilité) par rapport au taux de faux positifs (1 – taux de spécificité).

Si la courbe ROC coïncide avec la diagonale, le modèle est aussi performant qu'un modèle aléatoire où l'on attribue la classe au hasard. A l'inverse, plus la courbe ROC s'approche du coin supérieur gauche, meilleur est le modèle car il permet de capturer le plus possible de vrais positifs avec le moins possible de faux positifs.

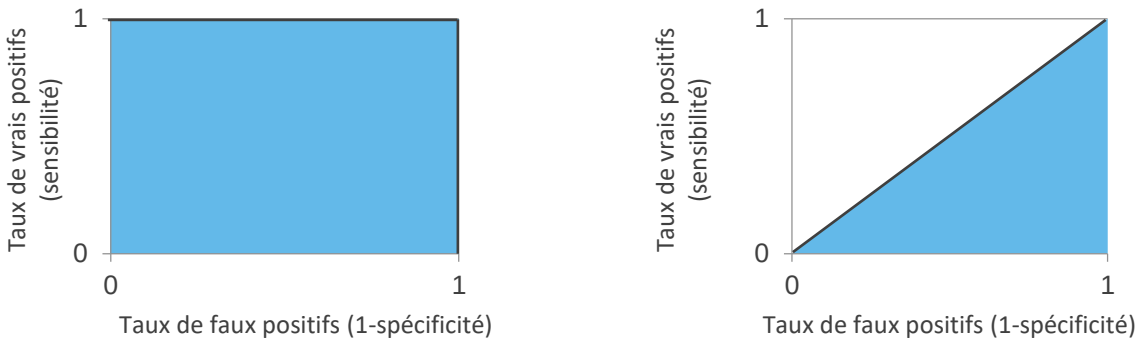


Figure 3.16 – Différences des courbes ROC pour un modèle optimal et un modèle aléatoire

3.5.2.8 AUC

L'AUC (ou Area under the ROC Curve) correspond à l'aire sous la courbe ROC. Une façon d'interpréter l'AUC est la probabilité que, parmi deux observations choisies au hasard, une observation de classe 1 et une autre de classe 0, la valeur du marqueur soit plus élevée pour l'observation de classe 1 que pour celle de classe 0. L'AUC varie de 0 à 1 et plus l'AUC est élevé, meilleur est le modèle car il a une forte capacité à distinguer les observations positives des observations négatives (ie prédire des classes 0 comme 0 et des classes 1 comme 1).

Dans le cas d'un modèle multiclasse, nous pouvons tracer une courbe ROC pour chaque classe en utilisant la méthode One VS All. Concrètement, si l'on dispose de trois classes nommées X, Y et Z, on tracera 3 courbes ROC :

- Une courbe pour X classé contre Y et Z
- Une autre courbe ROC pour Y classé contre X et Z
- Et une troisième courbe pour Z classé contre Y et X.

Les AUC de chaque classe individuelle sont calculées puis la moyenne de ces mesures permet d'obtenir l'AUC multiclasse :

$$AUC_{multi\ classe} = \frac{\sum_{i=1}^{nombre\ de\ classes} AUC_{classe\ n°i}}{nombre\ de\ classes}$$

3.5.3 Validation croisée et optimisation des hyperparamètres

Après avoir énuméré les différentes métriques de performance nécessaires à l'évaluation de la qualité d'une classification multiclasse, nous allons démarrer la construction de nos modèles. Pour ce faire, on réalise l'étape de séparation entre les observations d'entraînement et de test ainsi que la validation croisée. Elles visent à évaluer les performances de chaque modèle.

Lors de la première étape, le jeu de données prétraitées est divisé en deux parties : une partie test X_{test} et une partie entraînement X_{train}. La base de données train est utilisée pour la construction et l'apprentissage du modèle tandis que la partie test permet d'évaluer sa performance. Dans le cas de la classification en sévérité, nous divisons la base de données afin d'avoir 75% des données pour la partie train et 25% pour la partie test.

La seconde étape est celle de la validation croisée. Cette méthode vise à diviser une seconde fois la base de données en parties test et train. X_{train} est alors divisé en K sous-ensembles et l'objectif est d'utiliser un des K sous ensemble comme base de données test et les K-1 restants comme base de données train. Cette opération est répétée K fois pour chacun des K sous-ensembles disponibles pour le test. Le score de test final destiné à évaluer le modèle est la moyenne des performances de tous les tests effectués. Il existe une fonction Python qui utilise la méthode de validation croisée afin d'optimiser les hyperparamètres propres à chaque algorithme appelée GridSearchCV. Avec cette fonction, le calcul du score final est effectué pour chaque combinaison d'hyperparamètres du modèle. La combinaison d'hyperparamètres ayant obtenu le score le plus élevé est sélectionnée puis le modèle optimal est ré-entraîné une dernière fois sur toute la base de données X_{train}.

Dans le cadre de notre classification, nous optons pour une subdivision en K = 5 sous-ensembles. La métrique de performance utilisée pour le calcul du score final est le F1 score. Pour chaque modèle de machine learning, un ou plusieurs hyperparamètres seront testés. Dans le cadre de la vectorisation bag of words et TF IDF, le nombre de variables du modèle sera également optimisé. On considérera les n mots les plus fréquents dans la base de données d'entraînement dans le cas de bag of words et les n mots ayant la fréquence inverse la plus élevée dans le cadre de TF IDF. On testera 100, 1 000 et 5 400 variables.

Voici ci-dessous un schéma récapitulant les différentes étapes du processus de construction des modèles prédictifs.

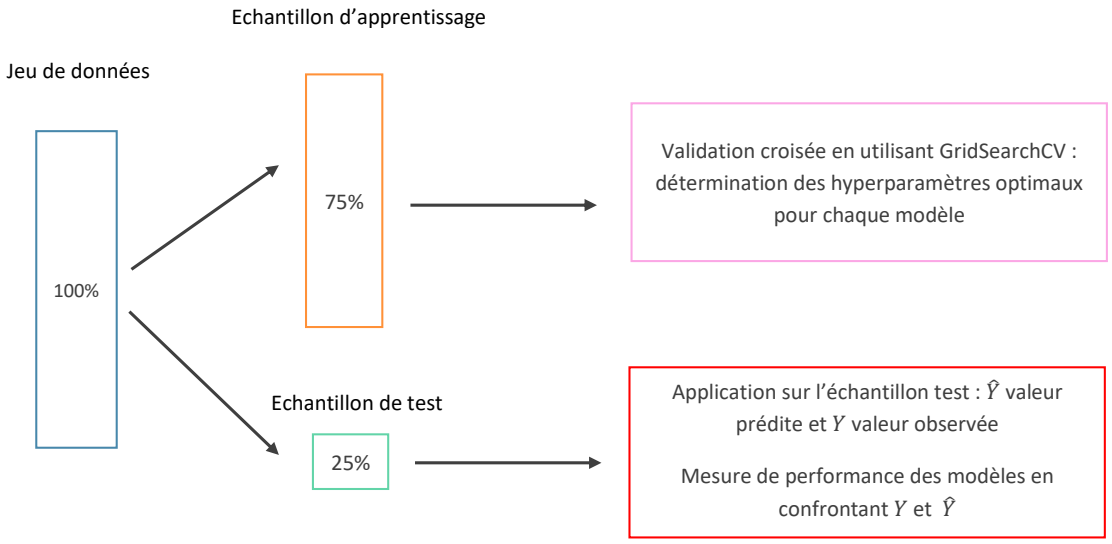


Figure 3.17 - Processus de construction des modèles de prédiction

Ce processus est réalisé trois fois : pour la vectorisation bag of words, TF IDF puis word2vec. Afin de comprendre plus en détails l'influence des hyperparamètres sur les performances des modèles, nous allons analyser ci-dessous les résultats de la méthode de validation croisée et nous allons nous focaliser sur le cas de la vectorisation TF IDF.

3.5.3.1 SVM

Le premier modèle testé est le modèle SVM. Nous avons sélectionné trois hyperparamètres à optimiser lors de la méthode de validation croisée :

- Le paramètre de régularisation C
- Le type SVM sélectionné
- Le nombre de variables sélectionnées lors de l'étape de vectorisation

Paramètre de régularisation :

Les paramètres de régularisation permettent de contrôler le surapprentissage dans un modèle de machine learning. Ils imposent une contrainte pour privilégier des modèles plus simples par rapport aux modèles complexes. Tout comme évoqué dans la partie 3.4.1, le paramètre de régularisation C dans le cas des SVM influence la taille de la marge de l'hyperplan de séparation. Une faible valeur de C implique une marge élevée et de nombreuses erreurs de classification tandis qu'un C plus élevé induit une marge moins importante et une fonction de décision plus complexe. Le réglage de ce paramètre permet ainsi de trouver un compromis entre la maximisation de la marge et le contrôle du taux d'erreurs.

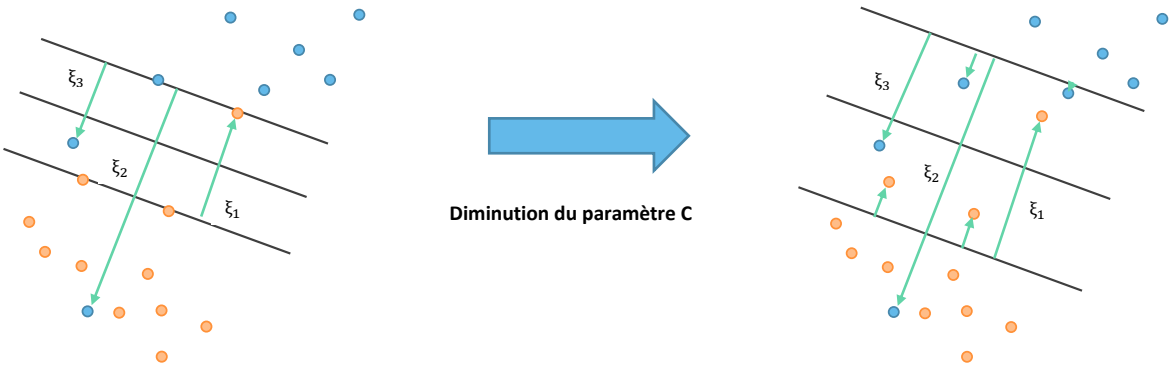


Figure 3.18 – Influence du paramètre C sur les erreurs de classification

Le type de SVM sélectionné :

Le second hyperparamètre optimisé est le type de SVM utilisé. Nous allons tester les SVM suivants :

- SVM linéaire
- SVM non linéaire à noyau polynomial
- SVM non linéaire à noyau gaussien
- SVM non linéaire à noyau tangente hyperbolique

La combinaison des hyperparamètres permettant de maximiser le F1 score est la suivante :

F1 score optimisé	C	Type de SVM	Nombre de variables
0.73	6	SVM linéaire (pas de fonction noyau)	5400

Tableau 3.6 – Combinaison optimale des hyperparamètres pour le SVM

Influence du type de SVM :

L'optimisation des hyperparamètres par la méthode de validation croisée indique que le SVM linéaire s'avère le plus efficace pour la classification en sévérité. En effet, contrairement aux autres problèmes de machine learning, le nombre de variables est particulièrement élevé pour la classification textuelle (ici le nombre de variables optimisé est de 5 400). Plus la dimensionnalité est élevée, plus il est facile de séparer linéairement les données et la projection des données dans un espace de dimension supérieure n'améliore pas les performances du modèle.

Influence du paramètre C :

Analysons l'influence du paramètre C sur les performances de notre algorithme. Le graphique suivant présente les résultats du F1 score du modèle en fonction de cet hyperparamètre (avec un SVM linéaire).

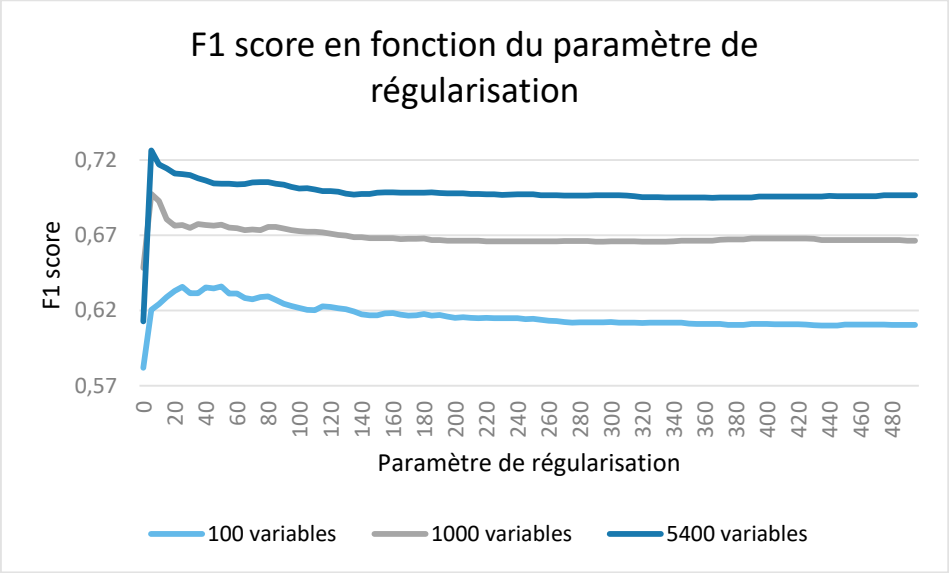


Figure 3.19 – F1 score en fonction du paramètre de régularisation

Premièrement, nous remarquons que la performance du modèle augmente avec le nombre de variables. Un nombre de variables significatif n'entraîne donc pas d'overfitting. Le F1 score augmente fortement jusqu'à C = 6 pour 5400 et 1000 variables et C = 35 pour 100 variables. Il se stabilise ensuite autour de 0.7 pour 5400 variables, 0.67 pour 1000 variables et 0.61 pour 100 variables. La complexification de la fonction de décision entraîne donc dans un premier temps une amélioration de la performance du modèle puis une diminution de cette dernière avec un phénomène de sur apprentissage.

3.5.3.2 Random Forest

Le second modèle testé dans la classification en sévérité est celui du random forest. Les paramètres influents sur la performance du modèle random forest sont nombreux.

Dans le cadre de notre classification, nous avons choisi d'optimiser les hyperparamètres suivants :

- La profondeur maximale de l'arbre : Il correspond au nombre maximal de niveaux des arbres construits à chaque itération. Plus l'arbre est profond et plus le nombre de variables sélectionnés pour la construction des arbres est élevé. La profondeur maximale de l'arbre variera dans l'intervalle $[\![5; 30]\!]$.
- Le nombre d'arbres utilisés pour la construction de la forêt qui variera dans l'intervalle $[\![5; 215]\!]$.
- Le nombre de variables sélectionnées lors de l'étape de vectorisation

La combinaison des hyperparamètres permettant de maximiser le F1 score est la suivante :

F1 score optimisé	Profondeur	Nombre d'arbres	Nombre de variables
0,67	26	160	1000

Tableau 3.7 – Combinaison optimale des hyperparamètres pour le random forest

Influence profondeur de l'arbre :

Analysons l'influence de la profondeur de l'arbre sur les performances de notre algorithme. Le graphique suivant présente les résultats F1 score de la validation croisée en fonction de la profondeur de l'arbre (avec un nombre d'arbres fixé à 160).

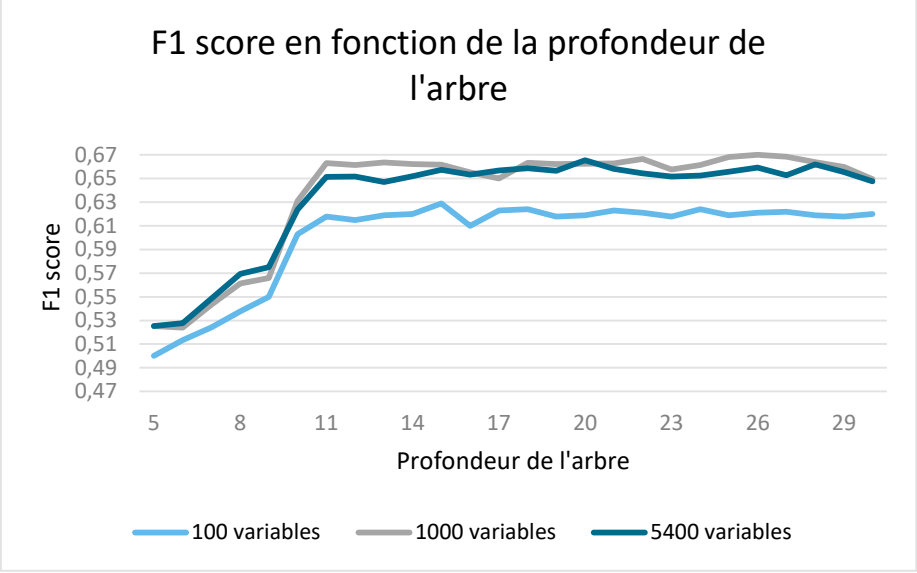


Figure 3.20 – F1 score en fonction de la profondeur de l'arbre

Nous remarquons tout d'abord que les performances du modèle avec 1000 et 5400 variables sont similaires. En enlevant les mots les moins pertinents de notre dictionnaire (avec un TF IDF faible), nous avons gardé une performance élevée et réduit significativement le nombre de variables du modèle. Le score est cependant moins élevé avec 100 variables.

Nous remarquons également que le score de nos trois modèles augmente avec la profondeur de l'arbre puis se stabilise autour de 0.66 (0.62 pour 100 variables) à partir d'une profondeur de 11. Une profondeur d'arbre trop faible entraîne un modèle trop simpliste et peut générer du sous apprentissage. Cependant, il est important de limiter la profondeur maximale car une valeur trop élevée de ce paramètre augmente la complexité du modèle et peut provoquer du sur apprentissage.

Influence du nombre d'arbres :

Le graphique suivant présente les résultats F1 score de la validation croisée en fonction du nombre d'arbres de la forêt (avec une profondeur maximale d'un arbre égale à 26).

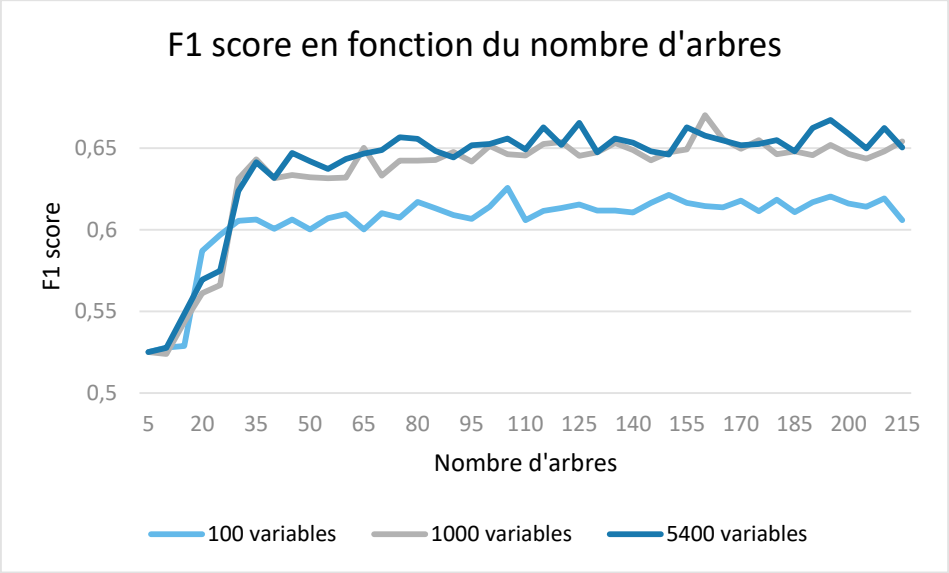


Figure 3.21 – F1 score en fonction du nombre d'arbres

Tout comme dans le graphique précédent, les performances du F1 score sont similaires pour 1 000 et 5 400 variables. Le F1 score augmente fortement pour un nombre d'arbres variant de 1 à 35 puis se stabilise à partir de 35 arbres. En règle générale, les performances d'un modèle random forest s'améliorent au fur et à mesure que le nombre d'arbres augmente, puis plafonnent à partir d'un certain seuil. Il est important de réguler ce paramètre car un modèle avec un nombre d'arbres élevé est coûteux en temps de calcul.

3.5.3.3 Gradient boosting

Le modèle de gradient boosting possède de nombreux paramètres communs au modèle de random forest tels que la profondeur de l'arbre ou bien le nombre d'arbres construits. Cependant, comme évoqué précédemment, les arbres sont créés de manière séquentielle, chaque arbre successif tentant de corriger les erreurs d'estimation des précédents. Le paramètre supplémentaire que nous allons considérer est le taux d'apprentissage qui mesure la contribution de chaque arbre dans le modèle.

Lors de l'étape de validation croisée, les hyperparamètres à optimiser sont les suivants :

- La profondeur maximale des arbres variant dans l'intervalle $[\![5\,;30]\!]$
- Le nombre d'arbres construits (c'est-à-dire le nombre d'itérations du modèle) variant dans l'intervalle $[\![5;215]\!]$
- Le taux d'apprentissage appartenant à $[0{,}1;0{,}6]$
- Le nombre de variables sélectionnées lors de l'étape de vectorisation

La combinaison des hyperparamètres permettant de maximiser le F1 score est la suivante :

F1 score optimisé	Profondeur	Taux d'apprentissage	Nombre d'arbres	Nombre de variables
0.68	15	0.1	128	1 000

Tableau 3.8 – Combinaison optimale des hyperparamètres pour le gradient boosting

Analysons l'influence du taux d'apprentissage sur la performance du modèle. Voici ci-dessous les résultats de F1 score du gradient boosting en fonction du taux d'apprentissage (avec une profondeur de 15 et un nombre d'arbres fixé à 128) :

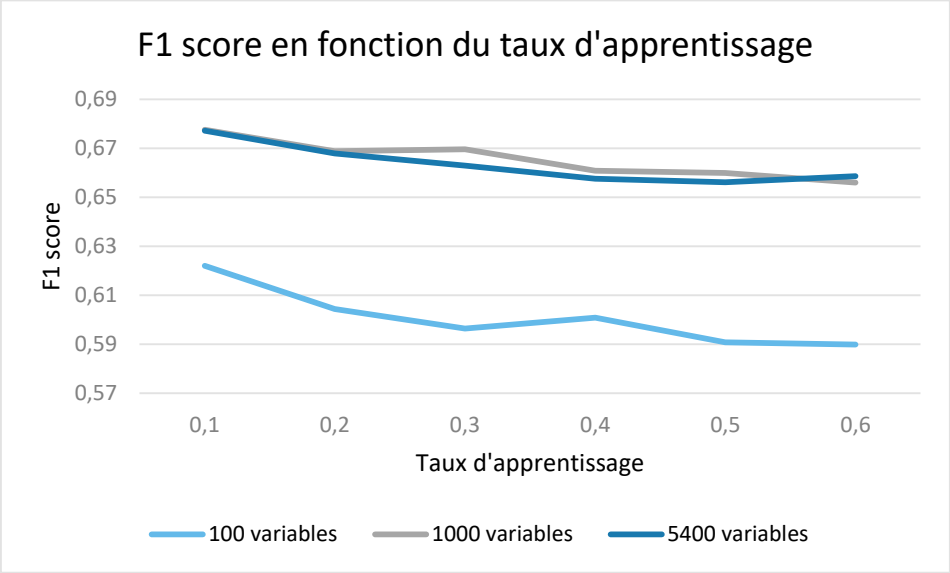


Figure 3.22 – F1 score en fonction du taux d'apprentissage

Nous remarquons que les performances avec 1 000 et 5 400 variables sont similaires. La performance du modèle est plus faible avec 100 variables. Dans les trois cas, l'augmentation du taux d'apprentissage provoque une diminution de la performance. En effet, un taux trop élevé empêche le modèle de converger vers un optimal et entraîne de l'underfitting. En revanche, avec un taux trop faible, la vitesse de convergence vers un optimal est lente.

3.5.3.4 Réseau de neurones

Le dernier modèle construit pour la classification en sévérité est le réseau de neurones. Les paramètres par défaut utilisés sont les suivants :

- La méthode d'ajustement des poids est celle de la descente de gradient
  La taille du batch égale à 200.
- Le nombre d'itérations : Nous l'initialisons à 200.
- Le taux d'apprentissage égal à 0.001
- La fonction d'activation des couches cachées : la fonction ReLU
- La fonction d'activation de la couche finale : la fonction softmax

Les hyperparamètres à optimiser dans la méthode de validation croisée sont les suivants :

- Le nombre de neurones : 2, 10, 50, 100, 150 puis 200 neurones par couches cachées.
- Le nombre de couches cachées variant de 1 à 5
- Le nombre de variables sélectionnées lors de l'étape de vectorisation

La combinaison des hyperparamètres permettant de maximiser le F1 score est la suivante :

F1 score optimisé	Nombre de neurones	Nombre de couches cachées	Nombre de variables
0.72	50	2	5 400

Tableau 3.9 – Combinaison optimale des hyperparamètres pour le réseau de neurones

Influence du nombre de neurones :

Sur le graphique ci-dessous, nous pouvons observer le F1 score obtenu en fonction du nombre de neurones des couches cachées (avec un nombre de couches cachées fixé à 2) :

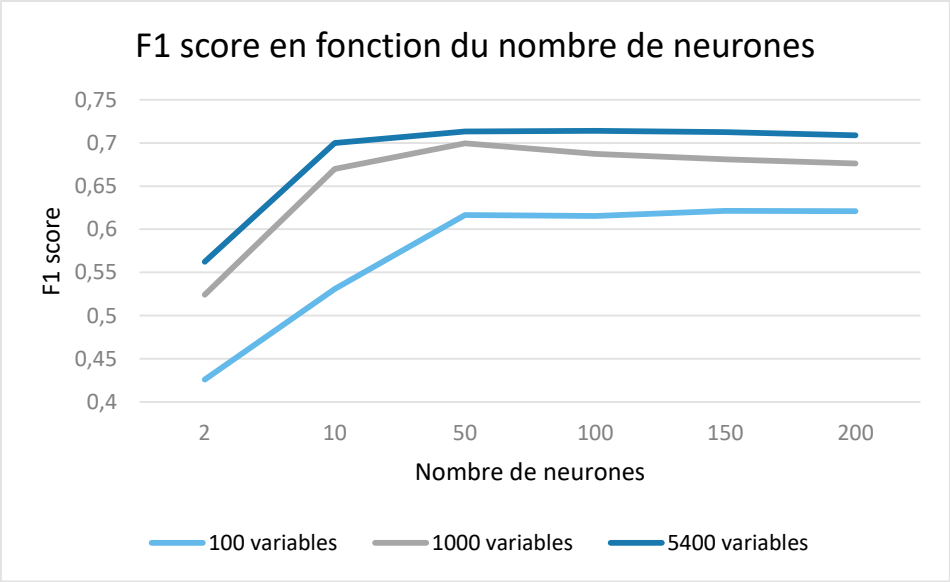


Figure 3.23 – F1 score en fonction du nombre de neurones

Premièrement, nous remarquons que la performance du modèle augmente avec le nombre de variables. Un nombre de variables significatif n'entraîne donc pas d'overfitting. De même, on constate une amélioration nette des performances lorsque le nombre de neurones augmente. Puis le F1 score se stabilise à partir de 50 neurones. Il est important de contrôler cet hyperparamètre car un nombre de neurones trop élevé peut entraîner un système trop complexe et peu adapté à notre base de données, provoquant ainsi un taux d'erreur trop important.

Influence du nombre de couches cachées :

Le graphique suivant présente les résultats F1 score de la validation croisée en fonction du nombre de couches cachées du réseau (avec 50 neurones sur chaque couche).

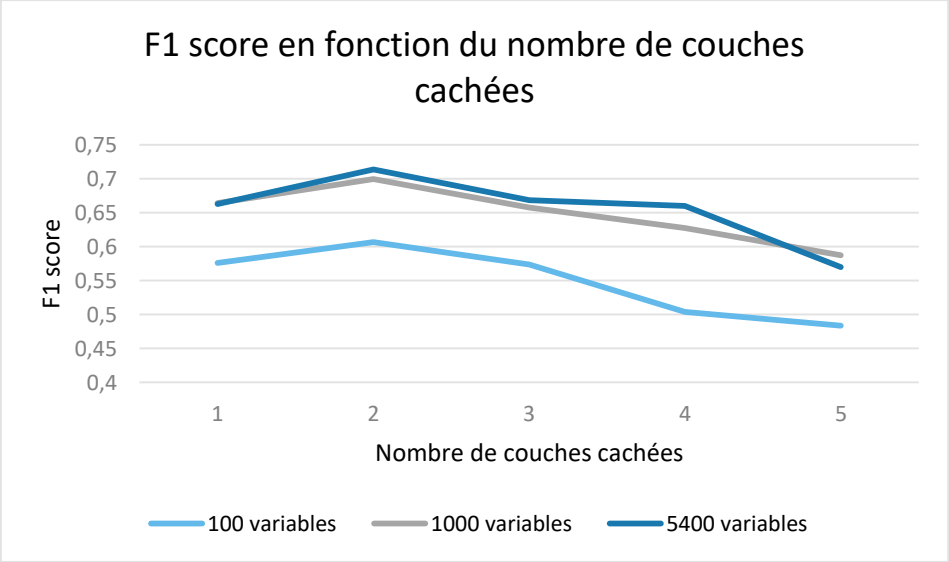


Figure 3.24 – F1 score en fonction du nombre de couches cachées

Nous remarquons tout d'abord que les performances du réseau avec 1 000 et 5 400 variables sont relativement proches. Pour 100, 1 000 et 5 400 variables sélectionnées, la performance augmente lorsque l'on passe d'une à deux couches cachées puis diminue. Tout comme pour le nombre de neurones, augmenter le nombre de couches cachées augmente la complexité du réseau, il est donc important de contrôler cet hyperparamètre afin d'éviter le surapprentissage.

3.5.3.5 Performances en fonction du nombre de variables

Dans les modèles analysés précédemment, nous avons remarqué que les performances différaient en fonction du nombre de variables utilisées. Certains modèles performaient davantage lorsque l'on sélectionnait 5400 variables tandis que d'autres obtenaient un F1 score équivalent pour 1000 et 5400 variables sélectionnées. Sur le graphique ci-dessous sont résumés les résultats de la méthode par validation croisée pour chacune des modélisations en fonction du nombre de variables.

	SVM	**Random Forest**	**Gradient Boosting**	**Réseaux de neurones**
100 variables	0,63	0,63	0,62	0,61
1 000 variables	0,70	0,67	0,68	0,70
5 400 variables	0,73	0,66	0,67	0,72

Tableau 3.10 –F1 score des modèles en fonction du nombre de variables (lors de la validation croisée avec une vectorisation en TF IDF)

Nous remarquons tout d'abord que les performances des modèles sont moindres lorsque le nombre de variables sélectionnées est égal à 100. Ensuite, pour chaque modèle, les F1 scores restent relativement proches lorsque l'on passe de 5 400 à 1 000 variables. En conservant les 1 000 mots les plus pertinents de notre dictionnaire (avec une pondération TF IDF élevée), nous avons conservé un score autour de 0,7 et réduit significativement le nombre de variables du modèle.

Nous sélectionnons pour la suite des analyses les modèles les plus performants selon la méthode de validation croisée. Ainsi, pour la vectorisation en TF IDF, on conserve :

- Le modèle à 5 400 variables pour l'algorithme SVM et le réseau de neurones.
- Le modèle à 1 000 variables pour les algorithmes de random forest et de gradient boosting.

3.5.4 Sélection de la méthode de vectorisation

L'optimisation des hyperparamètres de chaque algorithme fut également réalisée avec la vectorisation en bag of words et word2vec. Afin de sélectionner la méthode la plus efficace pour la classification en sévérité, nous prédisons les différentes classes sur la base de données X_{test} avec chacun des modèles optimisés. Nous obtenons les résultats ci-dessous :

	SVM	Random Forest	Gradient Boosting	Réseaux de neurones
Bag of words	0,44	0,45	0,41	0,48
TF IDF	0,72	0,71	0,70	0,66
Word2vec	0,24	0,56	0,53	0,55

Tableau 3.11 – F1 score des modèles en fonction de la vectorisation utilisée (sur la base de données test)

Nous remarquons que le F1 score diffère en fonction de la méthode de vectorisation utilisée. Tout d'abord, avec la vectorisation bag of words, le F1 score est relativement faible et ne dépasse pas les 50%. En effet, cette méthode offre une représentation trop naïve de l'ensemble du corpus et empêche d'obtenir des résultats performants. Pour ce qui est du modèle pré-entraîné par Google word2vec, les résultats varient en fonction de l'algorithme utilisé. La performance est considérablement réduite avec le modèle SVM. Pour ce qui est des modèles d'arbres et des réseaux de neurones, les résultats s'améliorent et atteignent environ 55%. Dans ce cas de classification, le modèle word2vec n'offre pas les résultats espérés, le sujet de l'assurance spatiale étant trop particulier. Pour la suite des analyses, nous sélectionnerons donc la vectorisation en TF IDF qui permet de rendre compte de l'importance des mots dans l'ensemble du corpus. Cette méthode offre de bien meilleurs résultats avec un F1 score dépassant 0.7 pour trois des modèles.

3.5.5 Comparaison des modèles

Concentrons-nous désormais sur les résultats des modèles de la vectorisation en TF IDF. Dans cette partie, nous allons évaluer la performance de chacun des modèles avec les métriques suivantes : l'accuracy, le F1 score et l'AUC.

	SVM	Random Forest	Gradient Boosting	Réseaux de neurones
Accuracy	0,74	0,72	0,71	0,68
F1 score	0,72	0,71	0,70	0,66
AUC	0,92	0,91	0,90	0,89

Tableau 3.12 – ou Métriques de performance des modèles (sur la base de données test)

Nous remarquons que les résultats des modèles sont relativement proches. L'accuracy et le F1 score tournent autour de 0.7 et l'AUC autour de 90%. Le réseau de neurones s'avère être le modèle le moins performant. Les résultats sont en revanche plus élevés pour le modèle SVM. Il sera donc sélectionné pour la suite des analyses.

3.5.6 Analyse du modèle sélectionné

Analysons plus en détail le modèle sélectionné. Dans le tableau ci-dessous sont présentées les mesures de précision, rappel et F1 score pour chacune des classes :

Étiquette	Précision	Rappel	F1 score	Nombre d'observations
Évènement sans impact	0,61	0,62	0,61	106
Perte de redondance ou marge dégradée	0,66	0,70	0,68	175
Perte partielle	0,78	0,75	0,77	273
Perte totale	0,82	0,81	0,82	252
Moyenne	**0,72**	**0,72**	**0,72**	**806**

Tableau 3.13 – Métriques de performance pour le modèle SVM sélectionné (sur la base de données test)

Nous remarquons que les performances diffèrent en fonction de la sévérité. Les anomalies en pertes totales et partielles obtiennent le F1 score le plus élevé. Les valeurs sont respectivement de 0,82 et 0,77. Les évènements sans impact et les pertes de redondance ont en revanche un F1 score inférieur à 70%. Ces deux classes obtiennent un score plus faible de précision. Cela indique un nombre plus élevé de faux positifs. Nous remarquons par exemple que sur tous les évènements classés sans impact par le modèle, seuls 61% le sont en réalité. De plus, le score de rappel obtenu indique que sur toutes les anomalies sans impact de la base de données, seules 62% ont été correctement classées par le modèle. Cette différence de performance peut être due au jeu de données déséquilibré. Le modèle a principalement appris les deux classes majoritaires de la base d'entraînement.

Toutes ces métriques calculées ont été obtenues grâce à la matrice de confusion ci-dessous. Elle compare les prévisions de chaque label aux valeurs réelles.

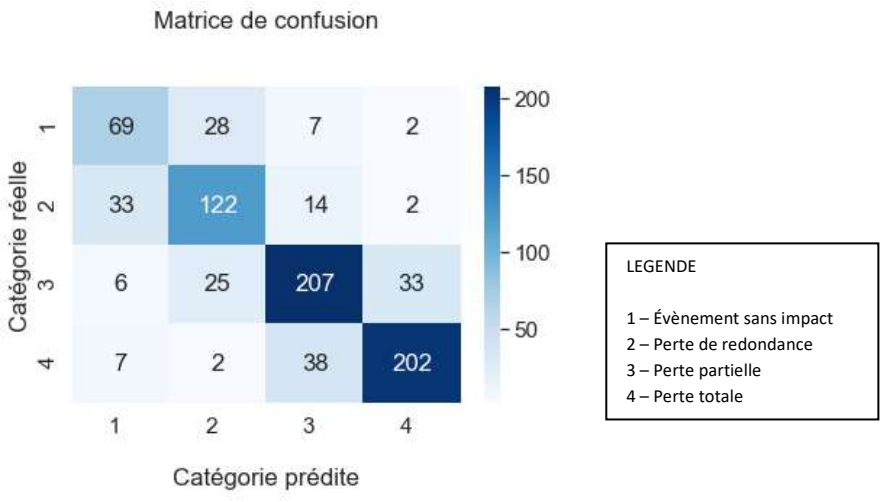


Figure 3.25 – Matrice de confusion du modèle SVM

Plus la diagonale de la matrice de confusion affiche une couleur foncée et meilleur est le modèle. Les cases foncées de la diagonale des anomalies « Perte partielle » et « Perte totale » indiquent une bonne performance des modèles pour ces deux classes.

En ce qui concerne l'étiquette prédite « Évènement sans impact », nous remarquons que les faux positifs sont en grande partie des anomalies qui auraient dû être classées en perte de redondance. De même, pour la catégorie réelle « Évènement sans impact », les faux négatifs sont principalement des anomalies qui ont été classées en perte de redondance.

Pour l'étiquette prédite « Perte de redondance », les faux positifs sont en grande partie des anomalies qui auraient dû être classées sans impact ou en perte partielle. Pour la catégorie réelle « Perte de redondance », les faux négatifs sont pour la plupart des anomalies qui ont été classées sans impact ou en perte partielle.

AUC et courbe ROC :

Un autre élément important à l'analyse des performances d'un modèle est l'AUC et la courbe ROC. Comme évoqué précédemment, la courbe ROC est le taux de vrais positifs (taux de sensibilité) par rapport au taux de faux positifs (1 – taux de spécificité).

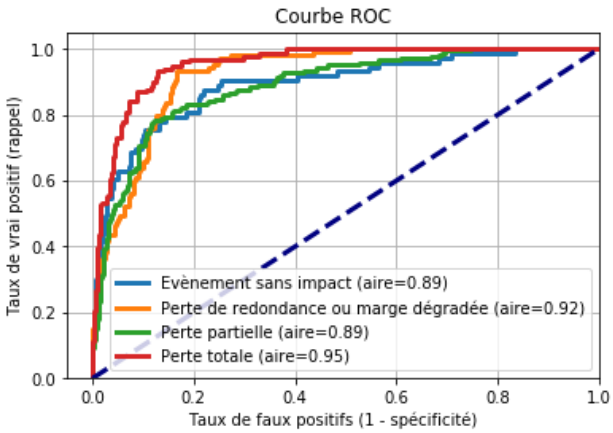


Figure 3.26 – Courbe ROC du modèle SVM

Nous remarquons ici que pour chacune des classes, le score d'AUC est élevé et tourne autour de 0,9 avec une courbe qui se rapproche du coin supérieur gauche. Cela signifie que le modèle a une forte capacité à distinguer les observations positives des observations négatives.

Nous remarquons cependant que les scores AUC sont beaucoup plus élevés pour chaque classe que les F1 scores. Ainsi, l'AUC totale qui correspond à la moyenne des AUC de chaque classe est de 0,91 contre 0,72 pour le F1 score. Le taux de faux positifs en abscisse de la courbe ROC est relativement stable lorsque le taux de négatifs est élevé. Le F1 score est donc une mesure à privilégier pour les jeux de données déséquilibrés, comme c'est le cas dans notre modélisation.

3.5.7 L'apprentissage semi-supervisé

Un des problèmes majeurs de la classification textuelle est l'acquisition des données étiquetées afin d'entraîner les modèles. Dans le cadre de l'étude des rapports de santé en assurance spatiale, obtenir des phrases d'anomalies étiquetées avec la sévérité correspondante est une démarche fastidieuse. Elle doit être faite à la main par une personne connaisseuse du domaine et peut prendre beaucoup de temps. Il existe cependant une autre méthode d'apprentissage en machine learning à mi-chemin entre l'apprentissage supervisé et non supervisé permettant de faire face à ce type de problème : la méthode semi-supervisée.

L'apprentissage semi-supervisé combine des exemples étiquetés et non étiquetés pour élargir l'ensemble de données disponibles pour la formation des modèles. En conséquence, nous pouvons améliorer les performances du modèle et économiser beaucoup de temps et d'argent en n'ayant pas à étiqueter manuellement de nombreux exemples.

Dans le cadre de notre étude, nous disposons de 2 403 données non étiquetées. Elles vont permettre d'enrichir la base de données d'entraînement pour la construction du modèle. L'ensemble des données utilisées et son découpage en base d'entraînement et de test sont présentés dans le schéma ci-dessous :

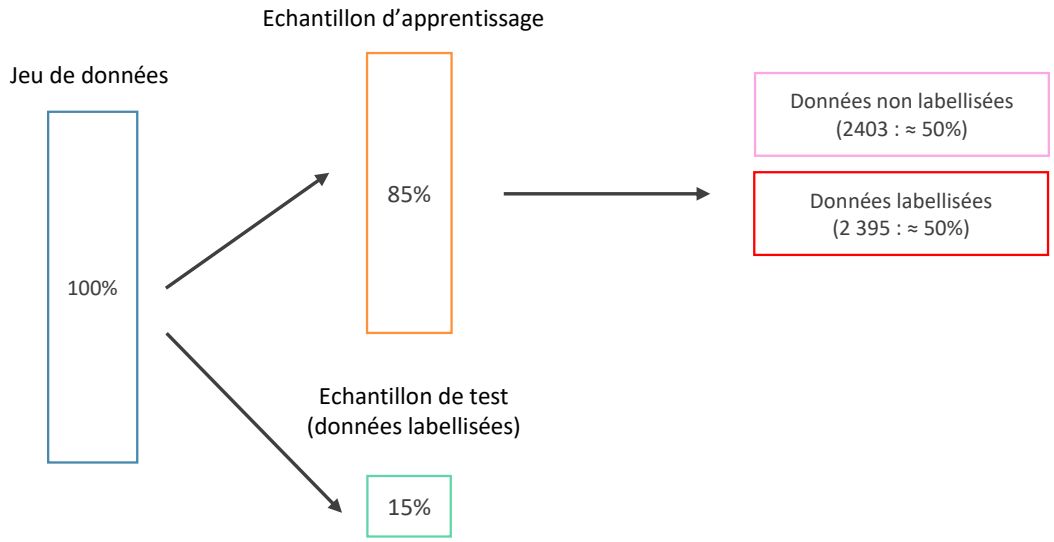


Figure 3.27 – Préparation des données pour l'apprentissage semi-supervisé

<u>Auto-apprentissage (ou self-training) :</u>

Une méthode couramment utilisée en apprentissage semi-supervisé est l'auto-apprentissage (ou self-training). Le principe de fonctionnement des algorithmes de self training est d'apprendre un classifieur de manière itérative en attribuant des pseudo-étiquettes à un ensemble des échantillons d'apprentissage non étiquetés. Ces exemples pseudo-étiquetés sont ensuite utilisés pour enrichir les données d'apprentissage étiquetées afin de former un nouveau classifieur. Les différentes étapes de l'algorithme sont résumées ci-dessous :

1) Tout d'abord, les données étiquetées sont utilisées pour former un premier modèle supervisé.
2) Ensuite, le modèle construit permet de prédire la classe des données non étiquetées.
3) Dans la troisième étape, les observations non étiquetées qui satisfont des critères prédéfinis sont sélectionnées. Il existe deux types de critères :
    - Méthode du seuil : Les observations dont la probabilité de prédiction est supérieure à un certain niveau p sont sélectionnées (par exemple, les observations dont la probabilité de prédiction est supérieure à 90 %).
    - Méthode *k_best* : Les k observations avec les probabilités de prédiction les plus élevées sont sélectionnées (par exemple, les 10 observations ayant les probabilités de prédiction les plus élevées).

    Ces pseudo-étiquettes sont ensuite combinées avec les données étiquetées.
4) Un nouveau modèle supervisé est formé en utilisant des observations avec les étiquettes et les pseudo-étiquettes. Grâce à ce nouveau modèle, les prédictions sont effectuées sur les données non étiquetées restantes et les observations nouvellement sélectionnées sont ajoutées à l'ensemble pseudo-étiqueté.
5) Les différentes étapes sont parcourues jusqu'à ce que toutes les données soient étiquetées, qu'aucune observation non étiquetée supplémentaire ne satisfasse le critère de pseudo-étiquetage ou bien que le nombre maximal d'itérations spécifié soit atteint.

Voici ci-dessous un schéma résumant toutes ces étapes :

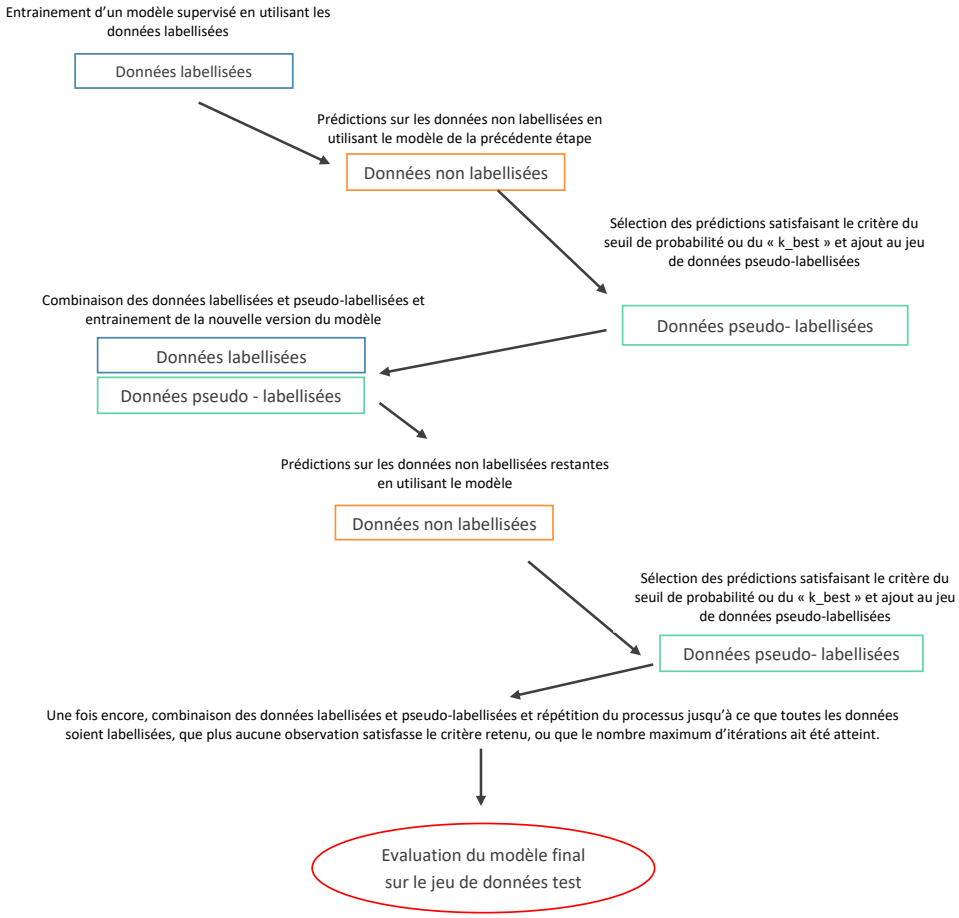


Figure 3.28 – Le processus itératif du self-training

Dans le cadre de ce mémoire, nous allons tenter d'améliorer le modèle SVM sélectionné dans la partie 3.5.6 en utilisant la méthode du self-training. Nous allons tester les deux critères de sélection de labels présentés ci-dessus : la méthode *k_best* et la méthode de seuil.

Méthode *k_best* :

Avec la méthode *k_best*, nous allons tester différentes valeurs de k en faisant varier ce paramètre de 20 à 55. En prédisant les classes sur notre échantillon test, nous obtenons les résultats de F1 score et AUC suivants :

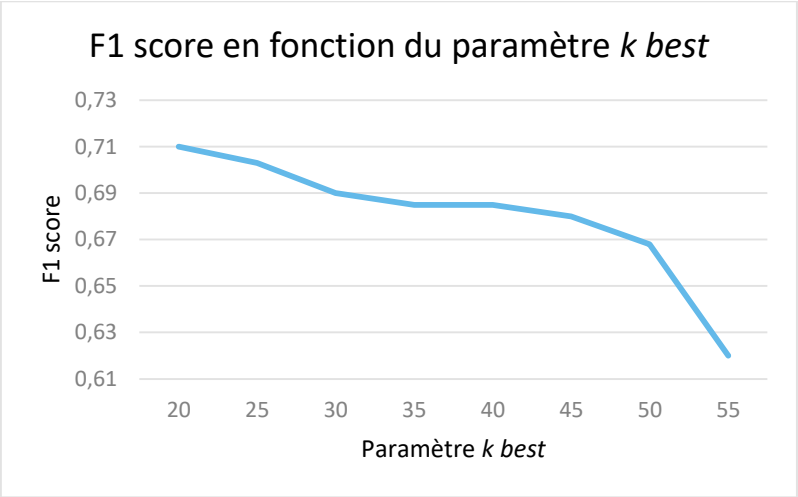


Figure 3.29 – F1 score en fonction du paramètre k_best

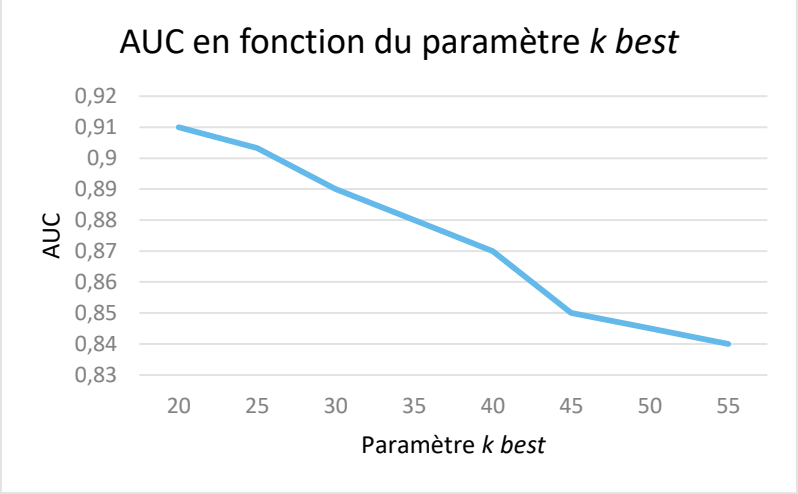


Figure 3.30 – AUC en fonction du paramètre k_best

Nous remarquons le F1 score et de l'AUC ont tendance à diminuer lorsque nous augmentons le paramètre k. En effet, avec un paramètre k élevé, nous avons plus de chances de sélectionner des données mal classifiées pour l'entraînement. La performance optimale est atteinte pour k = 20 avec un F1 score de 0,71 et un AUC de 0,91.

Méthode du seuil :

Avec la méthode du seuil, nous entraînons le modèle avec un seuil de probabilité p variant de 0,4 à 0,95. Le graphique ci-dessous présente le nombre de données pseudo non labellisées sélectionnées lors de la construction du modèle en fonction de la valeur de p.

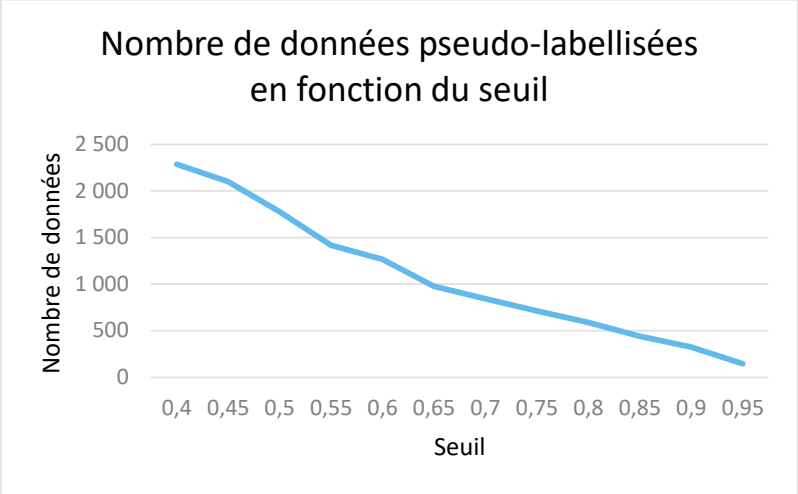


Figure 3.31 – Nombre de données pseudo-labellisées en fonction du seuil

Nous remarquons que pour des seuils de probabilité très élevés (dans l'intervalle $[0{,}9; 1[$), la quantité d'échantillons auto-étiquetés sélectionnés est faible. Le volume de données d'entraînement du classifieur n'augmente pas ou très peu. A l'inverse, pour un seuil bas (dans l'intervalle $[0{,}4; 0{,}5]$), la quantité d'échantillons auto-étiquetés sélectionnés est élevée et se rapproche de 2 200. On peut cependant retrouver dans ces données sélectionnées des échantillons à faible confiance qui ont probablement des étiquettes prédites incorrectes et, qui par conséquent, diminueraient la performance du classifieur.

Avec les modèles construits selon les différents seuils p, nous allons prédire les classes sur notre échantillon test. Nous obtenons les résultats de F1 score et AUC suivants :

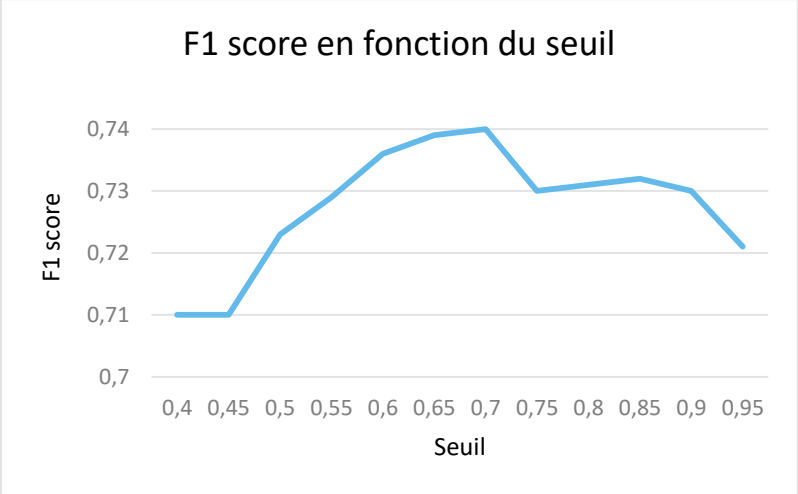


Figure 3.32 – F1 score en fonction du seuil

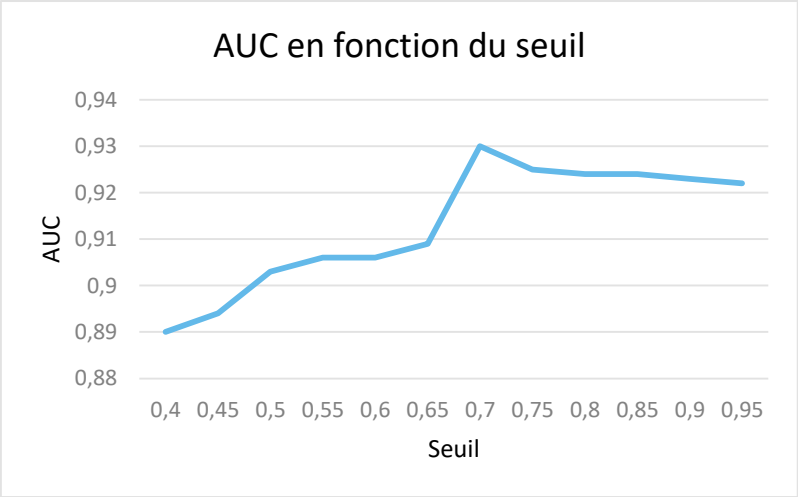


Figure 3.33 – AUC en fonction du seuil

Le seuil de probabilité optimal se situe à 0,7. Au-dessus de cette valeur, le modèle sélectionne peu de données pseudo-étiquetées à chaque itération et la performance finale obtenue se rapproche de la performance que l'on aurait obtenu sans self-training. En dessous de cette valeur, le classifieur apprend à partir d'échantillons étiquetés avec une faible confiance et qui ont alors probablement des étiquettes prédites incorrectes. Par conséquent, l'AUC et le F1 score sont moins élevés.

Modèles optimisés :

Les performances finales obtenues pour la méthode du seuil et la méthode *k_best* sont résumées ci-dessous :

Méthode	Paramètre optimal	F1 score	AUC
Méthode de seuil	seuil = 0,7	0,74	0,93
k_best	k = 20	0,71	0,91

Tableau 3.14 – Performances obtenues en fonction de la méthode utilisée

Selon les deux métriques, la méthode de seuil s'avère la plus performante. Cette méthode s'appuie sur les probabilités de prédiction pour sélectionner les observations à inclure dans le jeu de données d'entraînement tandis que la méthode *k_best* s'appuie sur les k observations ayant la probabilité estimée la plus élevée. Si l'ensemble des probabilités estimées sur le jeu de données est compris dans un intervalle à faibles valeurs, le k-échantillon pourrait inclure des observations avec une probabilité de prédiction inférieure à 0,5 et cela pourrait en conséquence diminuer la performance de notre modèle.

Analyse des performances de la méthode de seuil :

Nous sélectionnons la méthode de seuil pour la suite de nos analyses. Nous allons comparer la performance de ce modèle semi-supervisé avec celle obtenue avec le modèle supervisé. Le tableau

ci-dessous présente les résultats du F1 score, rappel et précision pour chacune des étiquettes avec le *self training classifier* construit :

Étiquette	Précision	Rappel	F1 score	Nombre d'observations
Évènement sans impact	0.65	0.67	0.66	106
Perte de redondance ou marge dégradée	0.71	0.72	0.71	175
Perte partielle	0.78	0.73	0.75	273
Perte totale	0.83	0.81	0.82	252
Moyenne	**0.74**	**0.73**	**0.74**	**806**

Tableau 3.15 – Performances obtenues pour la méthode de seuil

En comparant ce tableau avec les résultats obtenus en partie 3.5.6, nous remarquons que les performances de notre modèle se sont globalement améliorées en utilisant la méthode semi supervisée. Le F1 score est passé de 0.61 à 0.66 concernant les évènements sans impact, de 0.68 à 0.71 pour les évènements en perte de redondance et il est resté stable pour les pertes totales. En revanche, le F1 score des pertes partielles a diminué de 0.77 à 0.75. La méthode de self training a cependant amélioré la performance globale de notre modèle qui obtient une précision et un rappel de respectivement 0.74 et 0.73.

3.5.8 Conclusion sur les résultats obtenus

Dans le cadre de ce mémoire, différentes méthodes de traitement des données et de modélisation ont pu être testées. L'étape de vectorisation a permis de représenter numériquement les descriptions d'anomalies pour que les modèles de classification soient capables de les interpréter. Les analyses effectuées ont permis de conclure que la méthode de vectorisation TF IDF s'avérait être la plus performante pour notre jeu de données, la vectorisation bag of words offrant une représentation trop naïve de l'ensemble du corpus et le sujet du spatial étant trop particulier pour utiliser le modèle pré-entraîné word2vec.

Lors de l'étape de modélisation, quatre algorithmes ont été testés : les modèles SVM, random forest, gradient boosting et réseaux de neurones. Le paramétrage avec la méthode de validation croisée a permis de sélectionner la meilleure combinaison d'hyperparamètres pour chacun des modèles. La prédiction sur la base de données test et les diverses analyses effectuées ont permis d'apporter les conclusions suivantes :

- Les modèles optimisés possèdent des performances relativement proches avec un F1 score et une accuracy variant dans l'intervalle $[0{,}65; 0{,}75]$ et un AUC aux alentours de 0,9.

- Le modèle SVM s'avère cependant être le plus performant concernant la classification en sévérité. En effet, cet algorithme est très efficace dans des espaces à dimensions importantes et lorsque le nombre de variables est particulièrement élevé.

- Les modèles d'arbres sont légèrement moins performants. En effet, le nombre total de variables du jeu de données d'entraînement est relativement élevé et à chaque construction d'arbre, un nombre restreint de variables est sélectionné. L'algorithme peut ainsi

sélectionner des mots peu présents dans la base de données avec une pondération TF IDF nulle dans de nombreuses descriptions. Ces mots sont souvent peu significatifs pour la prédiction en sévérité. Le modèle SVM est ainsi plus adapté car elle considère l'ensemble des variables du corpus dans le processus de modélisation.

- Enfin, lors de l'application sur la base de données test, le réseau de neurones s'est révélé être le modèle le moins performant pour la classification en sévérité. L'algorithme se base sur une descente de gradient minimisant à chaque étape l'erreur sur l'ensemble des données d'apprentissage. Dans le cadre de notre étude, la base de données contient peu d'observations, de l'ordre de 3 000. Or les réseaux de neurones fournissent un apprentissage progressif à partir de gros volumes de données, il est donc difficile d'entraîner l'algorithme sur ce corpus.

Les algorithmes utilisés dans le cadre de ce mémoire sont difficiles à interpréter pour une classification textuelle. Ils agissent tels une « boîte noire » et n'expliquent pas ou très peu leurs décisions.

Après analyse du modèle SVM sélectionné, nous avons pu remarquer les limites de notre modélisation. Tout d'abord, le jeu de données d'entraînement est déséquilibré. Les valeurs de précision et de rappel obtenues pour les anomalies sans impact et en pertes de redondance sont moins élevées que pour les pertes partielles et totales. Ces écarts de performance sont dus à une sous-représentation de ces classes dans le jeu de données d'entraînement. De plus, le nombre d'observations utilisé pour l'apprentissage des modèles est relativement faible. Pour pallier ce problème, des méthodes d'apprentissage semi-supervisé ont été mis en œuvre dans la partie 3.5.7. La base de données d'entraînement fut enrichie de 2 403 données supplémentaires non labellisées. Deux méthodes ont été testées : la méthode *k_best* et la méthode de seuil. Après avoir déterminé la valeur de k et le seuil de probabilité p optimaux, les modèles ont été appliqués sur un jeu de données test. Nous avons pu remarquer que la méthode de seuil s'avérait être la plus performante.

Lors de la dernière étape des analyses, nous avons pu comparer les performances du modèle supervisé avec le modèle non supervisé. L'augmentation de la base de données d'entraînement à chaque itération avec des données pseudos labellisées a permis d'augmenter sensiblement les performances de notre algorithme SVM. Le F1 score final du modèle est ainsi passé de 0,72 à 0,74.

Conclusion

Les travaux réalisés dans le cadre de ce mémoire ont permis l'élaboration d'un outil d'extraction des anomalies présentes dans les bilans de santé des satellites, régulièrement envoyés par les opérateurs commerciaux à l'entreprise Scor. L'outil dispose d'une fonctionnalité visant à prédire la sévérité des anomalies extraites. Au cours de ce mémoire, les différentes étapes nécessaires à la construction de ce modèle de classification ont été décrites.

L'outil a pour objectif de faciliter le travail des souscripteurs en automatisant la tâche d'extraction des informations présentes dans les bilans de santé. Il sera capable d'extraire les principaux éléments issus des rapports et remplira automatiquement la base de données interne à l'entreprise Asterisk utilisée pour la souscription des contrats spatiaux.

L'outil final créé permet de traiter les bilans de santé de sept opérateurs différents. Il extrait les principaux éléments de chaque anomalie tels que la date de l'évènement, les équipements touchés ou encore le satellite concerné. L'outil possède cependant ses limites. Il nécessite des documents à la structure pérenne et qui proposent une certaine régularité dans les rapports. Il était donc impossible d'extraire automatiquement les informations pour tous les opérateurs assurés chez Scor car certains ne disposent pas de format normalisé et l'organisation des bilans peut changer d'une année à l'autre. De même, pour les opérateurs dont l'extraction des anomalies a pu être mise en place, il suffit que le nom d'une colonne d'un tableau ou que le titre d'une section soient modifiés pour empêcher la bonne extraction des informations. Ces défauts relevés par les souscripteurs permettront de solidifier la robustesse de l'outil au fur et à mesure de son utilisation.

Lors de mise en place des modèles de classification des anomalies, différentes étapes ont été réalisées. La première est la phase de préparation des données d'entraînement. Diverses méthodes de traitement de données ont ainsi été appliquées aux données textuelles brutes telles que le retrait des stop words, la racisation ou encore la vectorisation pour obtenir des données numériques.

Lors de l'analyse des performances des différents algorithmes entraînés, l'efficacité du modèle SVM a pu être constatée. Couramment utilisé dans les problèmes NLP, cet algorithme est particulièrement performant sur les données avec un nombre de variables particulièrement élevé, comme ce fut le cas dans notre modélisation. Lors de l'apprentissage des différents algorithmes, nous avons également pu remarquer l'une des limites des principaux projets de classification textuelle : le manque de données étiquetées pour l'apprentissage des algorithmes. Pour faire face à ce problème, une nouvelle méthode d'apprentissage a été testée : l'apprentissage semi-supervisé. En combinant les descriptions d'anomalies étiquetées avec des descriptions non étiquetées, la base de données a été enrichie et l'algorithme d'apprentissage semi-supervisé basé sur l'algorithme SVM a permis d'améliorer la performance du modèle.

Plus généralement, ce mémoire a mis en lumière les approches innovantes de la data-science pour le traitement des données textuelles. Dans le secteur de l'assurance qui dispose de grandes quantités d'informations non structurées, de plus en plus de projets de NLP voient le jour. Elles permettent notamment d'améliorer le processus de souscription comme dans le cadre de ce mémoire mais peuvent également être destinées à améliorer les services clients, la gestion des sinistres ou encore la détection des fraudes.

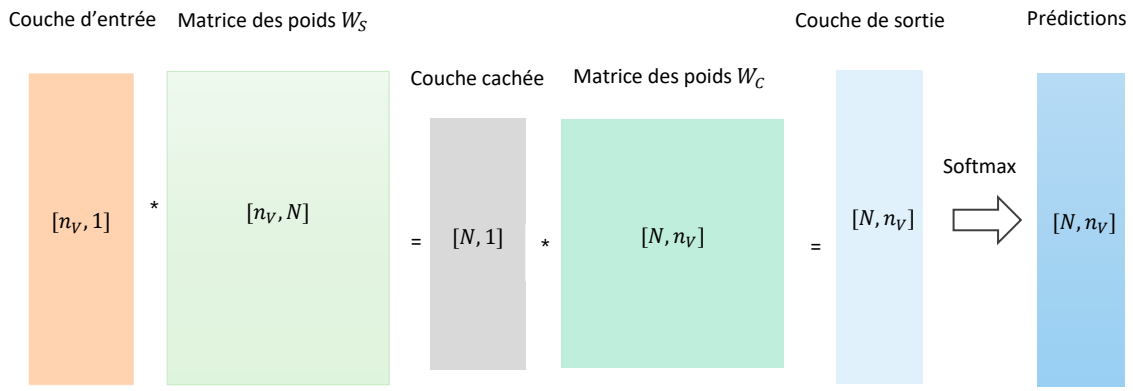
Bibliographie

[1] J.F. Gauché, « Space risk ». Master's thesis, centre d'études actuarielles (CEA), 2012.
[2] P. Manikowski and M.A. Weiss, « The Satellite Insurance Market and Underwriting Cycles », 2015.
[3] P. Montpert, « Cours sur l'assurance spatiale – ESTACA », 2015.
[4] Michal Zajac, « PANORAMA DE L'ASSURANCE SPATIALE », Scor
[5] C. Boyer, « Cours de Machine Learning », ISUP, 2021-2022.
[6] B. Pierrejean, « Qualitative Evaluation of Word Embeddings : Investigating the Instability in Neural - Based Models. », 2020
[7] A. Grünwald, « APPLICATIONS OF DEEP LEARNING IN TEXT CLASSIFICATION FOR HIGHLY MULTICLASS DATA », 2019
[8] Ajith Vajrala, « Text Classification », 2019
[9] WIENER M. et LIAW A., « Classification and Regression by randomForest », 2002
[10] XLStat, « Principe des courbes ROC », 2015
[11] Thomas G. DIETTERICH, « Ensemble Methods in Machine Learning », 2000
[12] Pradeep Kumar et Abdul Wahid, « Social Media Analysis for Sentiment Classification Using Gradient Boosting Machines », 2021
[13] Billal Belainine, « CLASSIFICATION SUPERVISÉE DE TEXTES COURTS ET BRUITÉS : APPLICATION AU DOMAINE DES MÉDIAS SOCIAUX », Université du Québec à Montréal
[14] Jean-Charles RISCH, « Enrichissement des Modèles de Classification de Textes Représentés par des Concepts », Université de Reims, 2017
[15] Taneli Saastamoinen, « Word2vec and its application to examining the changes in word contexts over time », Université d'Helsinki, 2020
[16] Isaac Haik, « Text mining et reconnaissance d'écriture appliqués à l'assurance », Institut des Actuaires, 2017
[17] Antoine Khan, « Analyse sémantique et prévention », Institut des Actuaires, 2019
[18] Yves Mercadier, « Classification automatique de textes par réseaux de neurones profonds : application au domaine de la santé », Université de Montpellier, 2020
[19] Céline Brouard, Thèse Inférence de réseaux d'interaction protéine-protéine par apprentissage statistique, 2013

Annexes

1 Algorithme de word2vec

Comme évoqué dans la partie 3.3.3 de ce mémoire, le modèle word2vec fonctionne sur le principe d'un réseau de neurones. Cet algorithme utilise un mot central pour prédire la probabilité que chaque mot du vocabulaire V de taille n_V, soit un mot de contexte dans une taille de fenêtre choisie. Le schéma du réseau est le suivant :



Fonctionnement du réseau de neurones word2vec

Le réseau prend en entrée un vecteur w_i associé au i-ème mot du vocabulaire. La première couche cachée renvoie un vecteur de dimension $(N, 1)$ avec N la dimension de projection choisie. Ce vecteur correspond au produit matriciel $W_S^T * w_i$. La dernière couche cachée correspond au produit matriciel $W_C * W_S^T * w_i$. La fonction softmax est ensuite appliquée sur les valeurs de sortie du réseau. Le i-ème élément du vecteur de sortie correspond à la probabilité que le i-ème mot du vocabulaire soit dans le contexte du mot correspondant au vecteur d'entrée.

Les étapes d'entraînement du réseau sont celles d'un réseau de neurones classique :

- Faire passer les données sur le réseau
- Évaluer l'erreur de la sortie (fonction de perte) en comparant les données réelles aux données prédites par le réseau
- Mettre à jour les poids dans le réseau correspondant aux matrices W_C et W_S en utilisant la méthode de descente de gradient pour réduire l'erreur
- Répéter jusqu'à ce que l'erreur soit minimisée

Pour comprendre plus en détails le fonctionnement de ce réseau, appuyons-nous sur un exemple : prenons la phrase "Frictions increase after launch and the increase indicates a problem". Prenons une fenêtre de taille 5 (c'est-à-dire pour le contexte deux mots avant et deux mots après). Les étapes d'entraînement du réseau sont les suivantes :

- Calculer la matrice de fréquence M telle que :
  $M[i, j] = fréquence\ du\ mot\ j\ dans\ le\ contexte\ du\ mot\ i$. M vaut ici :

	Frictions	Increase	After	Launch	And	The	Indicates	A	Problem
Frictions	0	0.5	0.5	0	0	0	0	0	0
Increase	0.14	0	0.14	0.14	0.14	0.14	0.14	0.14	0
After	0.25	0.25	0	0.25	0.25	0	0	0	0
Launch	0	0.25	0.25	0	0.25	0.25	0	0	0
And	0	0.25	0.25	0.25	0	0.25	0	0	0
The	0	0.25	0	0.25	0.25	0	0.25	0	0
Indicates	0	0.25	0	0	0	0.25	0	0.25	0.25
A	0	0.33	0	0	0	0	0.33	0	0.33
Problem	0	0	0	0	0	0	0.5	0.5	0

- Initialiser aléatoirement les matrices des poids W_c et W_S.
- A chaque itération, le réseau effectue les étapes suivantes :
  - Pour chaque mot w du vocabulaire :
    - Le réseau prend en entrée le vecteur one hot encoding associé au vecteur w_i. Pour le mot Increase par exemple, il s'agira du vecteur :

	Vecteur
Frictions	0
Increase	1
After	0
Launch	0
And	0
The	0
Indicates	0
A	0
Problem	0

    - Le vecteur de sortie du réseau est un vecteur de probabilités, par exemple :

	Vecteur
P(Frictions | increase)	0.12
P(Increase| increase)	0.04
P(After | increase)	0.24
P(Launch | increase)	0.18
P(And | increase)	0.15
P(The | increase)	0.05
P(Indicates | increase)	0.04
P(A| increase)	0.07
P(Problem | increase)	0.12

    - L'erreur associée au mot se calcule grâce à la matrice de fréquence M :

$$E_{thorn} = \frac{1}{2} * \sum_{j=1}^{n_V} (P(mot\ j \mid Thorn)_{réseau} - Freq(mot\ j \mid Thorn)_{mat\ M})^2$$

- Lorsque tous les mots sont passés dans le réseau, il suffit de calculer l'erreur totale de l'échantillon et d'ajuster les poids du réseau correspondant aux matrices W_c et W_S :

$$W_{t+1} = W_t - \alpha * \frac{1}{n_V} \sum_{i \in V} \frac{\partial E_i}{\partial W_t}$$

Après entraînement du réseau, à chaque mot du vocabulaire V est affecté un vecteur word embedding de dimension N. Il s'agit de la ligne de la matrice des sens W_s correspondant au mot considéré. Deux mots semblables avec un contexte similaire seront représentés par des vecteurs relativement peu distants.

2 Processus de tarification

Annexe confidentielle